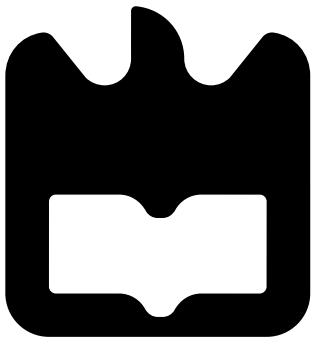
Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática,
2014

João Pedro
Ferreira Gonçalves

Sincronização de dois Receptores Heteródinos



João Pedro
Ferreira Gonçalves

Sincronização de dois Receptores Heteródinos

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor José Manuel Neto Vieira, Professor Associado com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

júri / the jury

presidente / president

Prof. Doutor Luis Filipe Mesquita Nero Moreira Alves
Professor Associado da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor José Manuel Neto Vieira
Professor Associado com Agregação da Universidade de Aveiro (Orientador)

Prof. Doutor Daniel Filipe Albuquerque
Professor Adjunto Convidado do Instituto Politécnico de Viseu - Escola Superior de Tecnologia de Viseu

Resumo

Joseph Mitola idealizou um *front-end* para Software Defined Radio (SDR), como sendo um rádio com capacidade de converter para o domínio digital sinais RF com uma largura de banda e gama dinâminca muito superiores aos permitidos pela actual tecnologia.
Analisando também o conceito de Cognitive-Radio, conclui-se que este tipo de rádio pressupõe a capacidade de analisar o espectro e poder determinar se existem bandas de frequência livres, isto é, que os proprietários dos direitos de transmissão nessa bandas não a estejam a usar num determinado instante. Para alcançar estes objectivos o rádio teria que ter a capacidade de usar um espectro vasto, composto por vários segmentos com diferentes larguras de banda e dispersos por uma grande banda espectral.
Um *front-end* habilitado a adquirir uma banda espectral muito elevada, que pode chegar facilmente às centenas de MHz ou até alguns GHz torna-se num problema bem complexo de resolver porque exige ao *front-end* ter uma frequência de amostragem e um número de bits muito superiores ao que é realizável nos dias de hoje.
Para efectuar a aquisição de um espectro vasto usando apenas um *front-end* é necessário que este opere a uma frequência de amostragem que seja, no mínimo, o dobro da largura de banda total do espectro. Esta abordagem torna-se pouco eficiente uma vez que obriga a digitalizar por inteiro o sinal com uma elevada largura de banda, quando na maioria das aplicações se pretende apenas aproveitar uma fracção desse espectro.
Neste trabalho é proposto uma abordagem diferente em que o receptor seria constituído por diversos *front-ends* heteródinos que convertem para o domínio digital as bandas dispersas ao longo do espectro. No caso de ser necessário adquirir um sinal com uma largura de banda superior à permitida por um só receptor irá-se usar dois *front-ends* que trabalhariam sincronizados de forma a adquirir um único sinal. Este problema de sincronização será o foco deste trabalho.

Abstract

The concept of Software Defined Radio (SDR) was idealized by Joseph Mitola as been a radio with capacity of conversion of RF signal to digital domain with a bandwidth and dynamic-range larger than what is achievable by actual technology.
Analyzing also the concept of Cognitive-Radio, it is concluded that this type of radio requires the ability to analyze the spectrum and determine if there are free frequency bands, ie, that the owners of the transmission rights of that band are not using it in a particular instant. To achieve these objectives the radio would have to have the ability to use a broad spectrum, consisting in several segments with different bandwidths and dispersed all-over a large spectral band.
A *front-end* able to acquire a very high spectral bandwidth, which can easily reach the hundreds of MHZ to a few GHz becomes in a complex problem to solve because it requires that *front-end* has a sampling frequency and a number of significants bits much higher than what is achievable today.
To make the acquisition of a wide range using a single *front-end* is necessary for this to operate at a sampling frequency that is at least twice of the total bandwidth of the spectrum. This approach becomes inefficient since it requires to converting the entire signal with high bandwidth, while in most applications we only want to utilize a fration of that spectrum.
This work proposes a different approach in which the receiver would consist of several heterodyne *front-ends* for converting to digital domain the frequency bands dispersed throughout the spectrum. Should it be necessary to acquire a signal with a bandwidth exceeding the permitted only by a receiver will use two *front-ends* that would work in synchronization in order to acquire a single signal. This synchronization problem will be the focus of this work.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ADC	Analog-Digital Converter
AFE	Analog Front-End
BPF	Filtro Passa-Banda
BPSK	Binary Phase Shift Keying
CR	Cognitive-Radio
CI	Circuitos Integrados
DAC	Digital-Analog Converter
dB	Decibel
DFE	Digital Front-End
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
FE	Front-End
FPGA	Full Programable Gates Array
FT	Fourier Transform
H/W	HardWare
IDFT	Inverse Discrete Fourier Transform
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
HFB	Hybrid Filter Bank
HPF	Filtro Passa-Alto

LIF	Low Intermediate-Frequency
LNA	Low Noise Amplifier
LPF	Filtro Passa-Baixo
PLL	Phase Lock-Loop
POCS	Projection On Convexs Sets
RF	Radio Frequency
SDR	Software Defined Radio
SNR	Signal-to-Noise Ratio
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral
WGN	White Gaussian Noise

Capítulo 1

Introdução

1.1 Enquadramento

Esta dissertação surge enquadrada no estudo sobre Cognitive-Radio (CR) onde se engloba o estudo de técnicas para a aquisição de sinais Radio Frequency (RF) de grande largura de banda. Devido às limitações de HardWare (H/W) e suas implementações surge a necessidade de descobrir novos métodos que permitam obter rádios com capacidade de tratar sinais de elevadas frequências (na ordem dos GHz) e larguras de banda consideráveis.

Perante esta realidade foi proposto o conceito de Software Defined Radio (SDR) (figura 1.1) apresentado pela primeira vez por Mitola [26], conceito que veio desencadear um grande número de estudos nessa área. O conceito baseia-se num rádio que é constituído por um bloco RF que recebe o sinal desejado, por um bloco que converte o sinal RF num sinal digital seguido por um Digital Signal Processor (DSP) que executa todo o processamento desejado no domínio digital. Esta arquitectura tem como principal objectivo obter radios dotados de uma grande capacidade de adaptação a vários cenários de transmissão e ao mesmo tempo apresentar níveis mínimos de interferência nos outros canais presentes no espectro electromagnético.

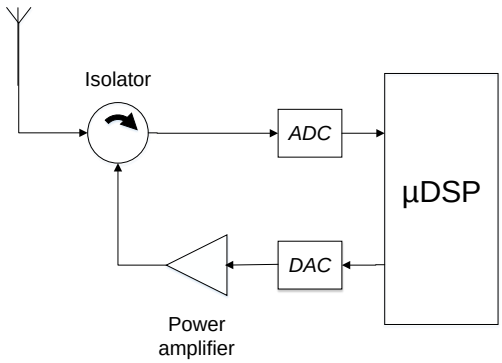


Figura 1.1: Arquitectura de um SDR

1.2 Objectivos

O objectivo principal desta dissertação é demonstrar que é possível ter dois andares super-heteródinos a trabalhar em modo cooperativo. Para alcançar este objectivo é necessário ter em consideração vários aspectos relativos com os componentes analógicos que constituem o Front-End, como as suas não-linearidades e imperfeições devido à implementação prática. Neste trabalho irá ser estudado o caso especifíco dos desvios de frequência e fase presente nos mixers e tentativa da sua compensação no domínio digital.

1.3 Estrutura da Dissertação

Esta dissertação será composta por 6 capítulos:

Capitulo 1 É feita uma breve descrição do problema tratado.

Capitulo 2 É realizada uma análise inicial ao estudo de dois andares super-heteródinos.

Capítulo 3 É estudado a teoria de Hybrid Filter Bank (HFB) e simulado o caso específico de dois canais.

Capítulo 4 É estudado o algortimo de Papoulis-Gerchberg que se inclui na teoria de Projection On Convexs Sets (POCS). Posteriormente este algoritmo é aplicado à teroria do capítulo 3.

Capítulo 5 Descrição da aquisição de sinais usando o Universal Software Radio Peripheral (USRP) e dos métodos de compensação de frequência e fase.

Capítulo 6 São levantados vários aspectos que podem ser melhorados nesta área e referenciadas algumas ideias que podem vir a ser concretizadas no futuro.

Capítulo 2

Caracterização de andares SuperHeteródinos

2.1 Front-End

Como foi introduzido anteriormente, o objectivo desta dissertação é a realização do estudo do funcionamento em cooperação de dois andares super-heteródinos. Esta tipologia de receptores tem a vantagem de converter o sinal RF para uma banda de frequência intermédia, o que permite a realização de filtros de baixa complexidade com uma selectivade considerável.

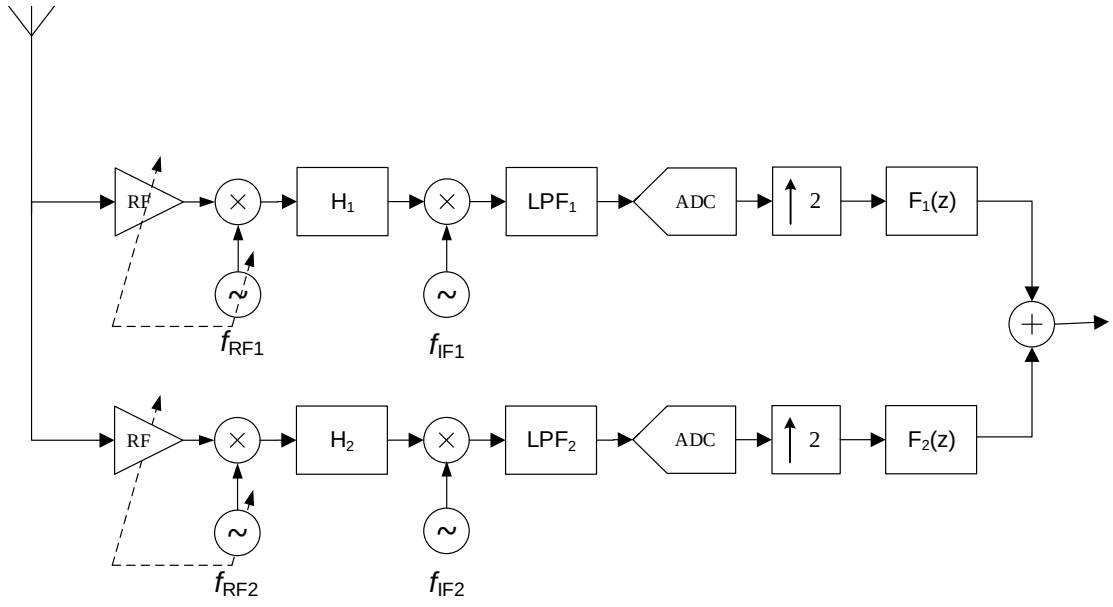
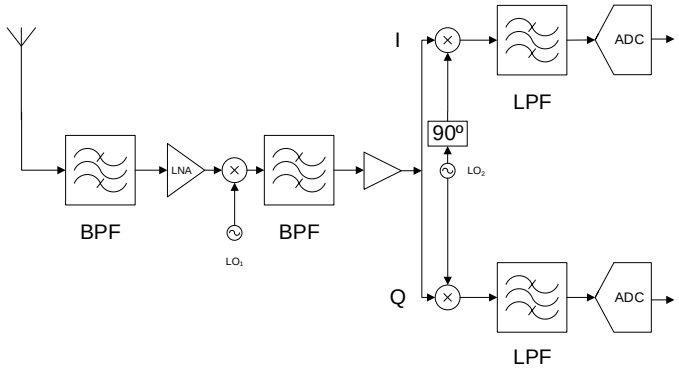


Figura 2.1: Receptor Super-Heteródino de dois canais

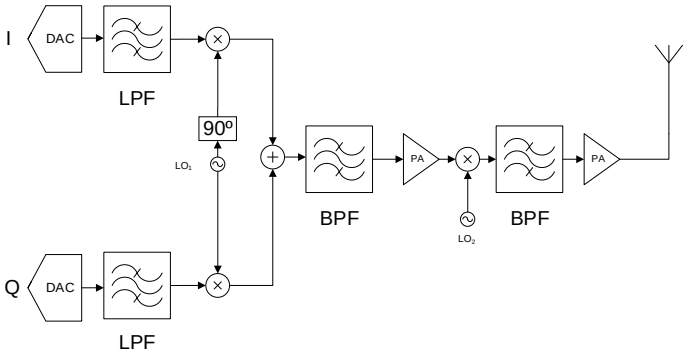
O modelo do sistema é apresentado na figura 2.1 que é composto pelo andar de Radio Frequency (RF), constituído pela antena, pelo pré-amplificador RF e pelo *mixer* de RF, seguido do andar Intermediate Frequency (IF), representado pelo filtro de análise em banda intermédia

e o *mixer* que converte o sinal IF para a banda-base e por último o andar em banda base que converte o sinal analógico num sinal digital através das Analog-Digital Converters (ADCs) antecidida por um filtro passa-baixo para eleminar os canais adjacentes. Após a digitalização os sinais são combinado num só por forma a poderem ser procesados por um Digital Signal Processor (DSP). Tendo em conta que todos estes componentes referidos apresentam factores não-ideais que introduzem vários tipos de erro é necessário compensá-los, sendo esta compensação realizada no domínio digital.

Os receptores e os transmissores (figura 2.2) podem ser usados como Digital Front-End (DFE) apesar da sua dificil integração em Circuitos Integrados (CI) [5]. O termo Front-End surge na sequência de uma pequena diferença em relação ao Software Defined Radio (SDR) ideal, uma vez que é necessário converter as caracteristicas do sinal RF de modo a ser possível realizar a sua digitalização para poder ser processado pelo DSP de acordo com os objectivos do rádio em questão. O FE é composto por uma parte analógica, Analog Front-End (AFE), e uma parte digital, DFE. Por outras palavras, um Front-End tem como função fazer a ponte de ligação entre o sinal RF captado pela antena e o sinal entregue ao DSP.



(a) Receptor Super-Heteródino



(b) Transmissor Super-Heteródino

Figura 2.2: Receptor e transmissor Super-Heteródino com modulação IQ

Na figura 2.2 está representado o receptor e o transmissor do tipo super-heteródino com modulação IQ. Esta modulação é usada para o tratamento de sinais complexos, isto é, que não possuam simetria par ao longo do eixo das frequências. Devido à modulação IQ é necessário

trabalhar com filtros complexos, um filtro para a componentes I (*In Phase*) e outro para a componente Q (*Quadrature*)

Existem duas abordagens para a representação de sinais complexos [32]: a primeira usando as componentes *InPhase* e *Quadrature* e a segunda usando a representação analítica. A primeira pode ser entendida como uma modelação de um sinal real por uma portadora complexa de frequência ω_c [13], representando no domínio da frequência uma tranlação de ω_c. No segundo caso, as componentes de frequências negativas são consideradas nulas.

A realização de filtros complexos pode-se realizar através de translações lineares da forma:

$$z^{-1} \rightarrow z^{-1}e^{j\omega} = z^{-1}(cos(\omega) + jsin(\omega)) \tag{2.1}$$

$$H_{real}(z) \rightarrow H_{complex}(z) = H_R(z) + jH_I \tag{2.2}$$

Exemplificando, aplicando a transformação linear da equação 2.1 a um filtro real passa-baixo, é possível obter um filtro complexo passa-banda que apresenta simetria em torno da sua frequência central. A função transferência do filtro complexo obtido apresenta coeficientes complexos e a mesma ordem do filtro protótipo real.

Tendo em consideração que:

$$Y(z) = H_{complex}(z)X(z) \tag{2.3}$$

e que ambos os termos são complexos na forma:

$$Y(z) = Y_R(z) + jY_I; \quad X(z) = X_R(z) + jX_I; \quad H(z) = H_R(z) + jH_I \tag{2.4}$$

é possível processar facilmente sinais complexos usando circuitos digitais complexos [31].

O sinal $Y(z)$ fica na forma:

$$\begin{aligned} Y(z) &= [H_R(z) + jH_I][X_R(z) + jX_I] \\ &= [H_R(z)X_R(z) - H_I(z)X_I(z)] + j[H_I(z)X_R(z) + H_R(z)X_I(z)] \end{aligned} \tag{2.5}$$

obtendo-se o diagrama de blocos de um filtro complexo:

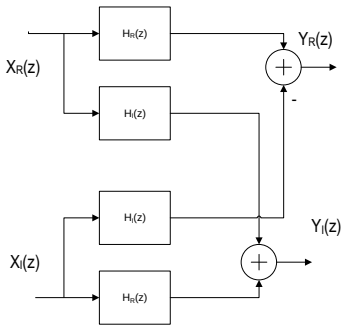


Figura 2.3: Diagrama de blocos de um filtro complexo

2.2 Modelo

Para uma melhor descrição do problema estudado consideramos um sinal em banda base com uma largura de banda $2 * BW$ (figura 2.4) em que BW representa a largura de banda de cada receptor heteródino.

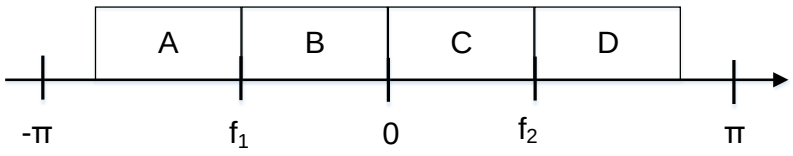


Figura 2.4: Espectro da frequência do sinal de entrada

Supondo que o sinal de entrada é dado pela forma:

$$x(t) = A_A \cos(2\pi F_A t) + A_B \cos(2\pi F_B t) + A_C \cos(2\pi F_C t) + A_D \cos(2\pi F_D t) \tag{2.6}$$

Fazendo uso da relação de *Euler*:

$$cosx = \frac{1}{2}(e^{jx} + e^{-jx}) \tag{2.7}$$

a equação 2.6 fica na forma:

$$x(t) = \frac{1}{2}\Big(A_A[e^{j2\pi F_A t} + e^{-j2\pi F_A t}] + A_B[e^{j2\pi F_B t} + e^{-j2\pi F_B t}] + A_C[e^{j2\pi F_C t} + e^{-j2\pi F_C t}] + A_D[e^{j2\pi F_D t} + e^{-j2\pi F_D t}]\Big) \tag{2.8}$$

Em que F_A, F_B, F_C, e F_D representam as frequências centrais das fracções A, B, C e D respectivamente. O mesmo se aplica A_A, A_B, A_C e A_D que representam as diferentes amplitudes.

Para se conseguir adquirir todo o sinal desejado é necessário realizar a análise do sinal de entrada, isto é, dividi-lo em dois sinais distintios com metade da largura de banda original. Esta análise é feita através do deslocamento da fracção A e B do sinal que se encontra centrado em f_1 para uma frequência central igual a 0 seguido de um filtro passa-baixo H_1 que elimina as compenentes não pertencentes ao canal. O mesmo é aplicado à fracção C e D que estão centradas em f_2. O diagrama de blocos desta operação pode ser observado na figura 2.5:

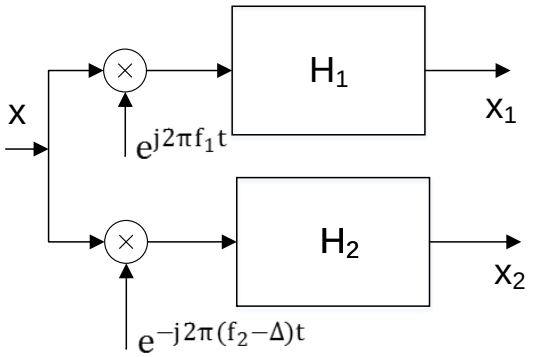


Figura 2.5: Diagrama de blocos da operação de análise

Aplicar os *mixers* no sinal de entrada equivalente a multiplicar a equação 2.6 por:

$$x_1(t) = x(t) \times e^{j2\pi f_1 t} \tag{2.9}$$

e

$$x_2(t) = x(t) \times e^{-j2\pi(f_2-\Delta)t} \tag{2.10}$$

Resultando os sinais $x_1(t)$:

$$\begin{aligned} x_1(t) = \frac{1}{2}\Big[& A_A(e^{j2\pi(F_A+f_1)t} + e^{-j2\pi(F_A-f_1)t}) + A_B(e^{j2\pi(F_B+f_1)t} + e^{-j2\pi(F_B-f_1)t}) + \\ & A_C(e^{j2\pi(F_C+f_1)t} + e^{-j2\pi(F_C-f_1)t}) + A_D(e^{j2\pi(F_D+f_1)t} + e^{-j2\pi(F_D-f_1)t})\Big] \end{aligned} \tag{2.11}$$

onde as componentes refentes às frequências F_C e F_D são atenuadas pelo filtro H_1 e do segundo canal resulta $x_2(t)$:

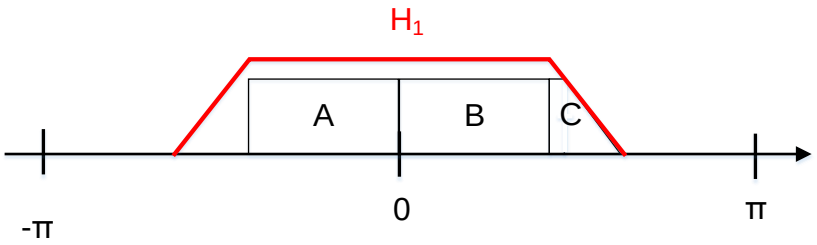
$$\begin{aligned} x_2(t) = \frac{1}{2}\Big[& A_A(e^{j2\pi(F_A-(f_2-\Delta))t} + e^{-j2\pi(F_A+(f_2-\Delta))t}) + A_B(e^{j2\pi(F_B-(f_2-\Delta))t} + e^{-j2\pi(F_B+(f_2-\Delta))t}) + \\ & A_C(e^{j2\pi(F_C-(f_2-\Delta))t} + e^{-j2\pi(F_C+(f_2-\Delta))t}) + A_D(e^{j2\pi(F_D-(f_2-\Delta))t} + e^{-j2\pi(F_D+(f_2-\Delta))t})\Big] \end{aligned} \tag{2.12}$$

o que resulta em:

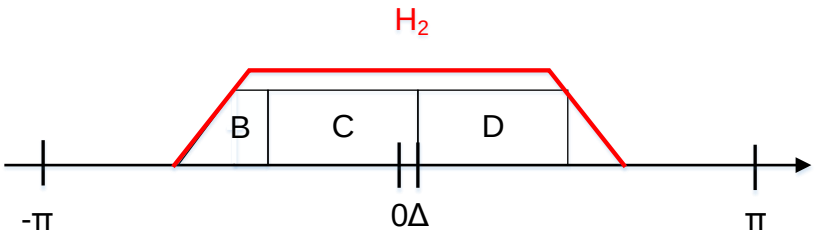
$$x_2(t) = \frac{1}{2}\Big[A_A(e^{j2\pi(F_A-f_2+\Delta)t} + e^{-j2\pi(F_A+f_2-\Delta)t}) + A_B(e^{j2\pi(F_B-f_2+\Delta)t} + e^{-j2\pi(F_B+f_2-\Delta)t}) + A_C(e^{j2\pi(F_C-f_2+\Delta)t} + e^{-j2\pi(F_C+f_2-\Delta)t}) + A_D(e^{j2\pi(F_D-f_2+\Delta)t} + e^{-j2\pi(F_D+f_2-\Delta)t})\Big] \tag{2.13}$$

onde são as componentes F_A e F_B que são afectadas pelo filtro H_2. O factor Δ que se observa no *mixer* do segundo canal representa o erro existente entre os dois osciladores sinusódais f_1 e f_2.

Os espectros obtidos após a análise são representados nas figuras 2.6 onde se verifica o deslocamento na frequência e a atenuação das componentes adjacentes ao sinal. O espectro do sinal x_1 contém também uma parte da fração C devido aos filtros não serem ideais e não possuirem uma banda de rejeição infinita. O mesmo acontece no espectro x_2 sendo a fracção B que se encontra atenuada pelo filtro H_2.



(a) Espectro da frequência do sinal X_1



(b) Espectro da frequência do sinal X_2

Figura 2.6: Espectro da frequência dos dois andares de análise

Nesta altura pode-se processar à digitalização dos dois sinais usando duas ADCs independentes a operarem a uma frequência de amostragem que seja metade da que era necessária no caso de se usar apenas uma. Já no domínio digital é necessário efectuar a síntese (figura 2.7)dos sinais obtidos à saída das ADCs para este poder ser processado por um DSP. Basicamente este processo realiza-se atráves do deslocamento dos dois sinais que se encontram em

torno de 0 para as suas respectivas frequências f_1 e f_2 e a sua soma. Desta forma obtém-se o sinal de entrada original.

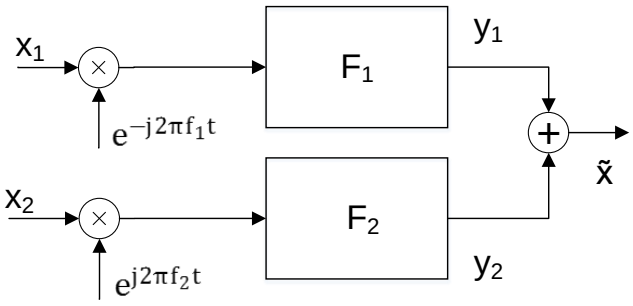


Figura 2.7: Diagrama de blocos da operação de síntese

Aplicando os *mixers* do bloco de síntese são obtidos os sinais y_1 e y_2.

$$y_1 = x_1 \times e^{-j2\pi f_1 t} \tag{2.14}$$

que resulta em:

$$y_1 = \frac{1}{2}\Big[A_A(e^{j2\pi(F_A+f_1)t} + e^{-j2\pi(F_A-f_1)t})e^{-j2\pi f_1 t} + A_B(e^{j2\pi(F_B+f_1)t} + e^{-j2\pi(F_B-f_1)t})e^{-j2\pi f_1 t} + A_C(e^{j2\pi(F_C+f_1)t} + e^{-j2\pi(F_C-f_1)t})e^{-j2\pi f_1 t} + A_D(e^{j2\pi(F_D+f_1)t} + e^{-j2\pi(F_D-f_1)t})e^{-j2\pi f_1 t}\Big] \tag{2.15}$$

apresentando em relação a x, apenas diferenças nas amplitudes alteradas pelo filtro H_1. No caso de y_2:

$$y_2 = x_2 \times e^{j2\pi f_2 t} \tag{2.16}$$

que resulta em:

$$y_2 = \frac{1}{2}\Big[A_A(e^{j2\pi(F_A-f_2+\Delta)t} + e^{-j2\pi(F_A+f_2-\Delta)t})e^{j2\pi f_2 t} + A_B(e^{j2\pi(F_B-f_2+\Delta)t} + e^{-j2\pi(F_B+f_2-\Delta)t})e^{j2\pi f_2 t} + A_C(e^{j2\pi(F_C-f_2+\Delta)t} + e^{-j2\pi(F_C+f_2-\Delta)t})e^{j2\pi f_2 t} + A_D(e^{j2\pi(F_D-f_2+\Delta)t} + e^{-j2\pi(F_D+f_2-\Delta)t})e^{j2\pi f_2 t}\Big] \tag{2.17}$$

quando simplificado, obtém-se o sinal $y_2(t)$ na forma:

$$y_2 = \frac{1}{2}\Big[A_A(e^{j2\pi(F_A+\Delta)t} + e^{-j2\pi(F_A-\Delta)t}) + A_B(e^{j2\pi(F_B+\Delta)t} + e^{-j2\pi(F_B-\Delta)t}) + A_C(e^{j2\pi(F_C+\Delta)t} + e^{-j2\pi(F_C-\Delta)t}) + A_D(e^{j2\pi(F_D+\Delta)t} + e^{-j2\pi(F_D-\Delta)t})\Big] \tag{2.18}$$

Pela observação do sinal reconstruído verifica-se que na banda de adjacência dos dois canais aparecem componentes que não estavam presentes no sinal de entrada. Este acontecimento deve-se ao deslocamento na frequência provocado pelo factor Δ:

$$\begin{aligned}\tilde{x} &= y_1 + y_2 \\ &= \frac{1}{2}\Big[A_A(e^{j2\pi F_A t} + e^{-j2\pi F_A t} + e^{j2\pi(F_A+\Delta)t} + {}^{-j2\pi(F_A-\Delta)t}) + \\ &\quad A_B(e^{j2\pi F_B t} + e^{-j2\pi F_B t} + e^{j2\pi(F_B+\Delta)t} + {}^{-j2\pi(F_B-\Delta)t}) + \\ &\quad A_C(e^{j2\pi F_C t} + e^{-j2\pi F_C t} + e^{j2\pi(F_C+\Delta)t} + {}^{-j2\pi(F_C-\Delta)t}) + \\ &\quad A_D(e^{j2\pi F_D t} + e^{-j2\pi F_D t} + e^{j2\pi(F_D+\Delta)t} + {}^{-j2\pi(F_D-\Delta)t})\Big]\end{aligned} \tag{2.19}$$

Tendo em conta os efeitos provocados pelo filtros de análise H_1 e H_2, que podem ser observados na figura 2.8, a equação 2.19 fica na forma:

$$\begin{aligned}\tilde{x} &= y_1 + y_2 \\ &= \frac{1}{2}\Big[A_A(e^{j2\pi F_A t}) + A_B(e^{j2\pi F_B t} + e^{-j2\pi F_B t} + e^{j2\pi(F_B+\Delta)t} + {}^{-j2\pi(F_B-\Delta)t}) + \\ &\quad A_C(e^{j2\pi F_C t} + e^{-j2\pi F_C t} + e^{j2\pi(F_C+\Delta)t} + {}^{-j2\pi(F_C-\Delta)t}) + A_D(e^{j2\pi(F_D+\Delta)t} + {}^{-j2\pi(F_D-\Delta)t})\Big]\end{aligned} \tag{2.20}$$

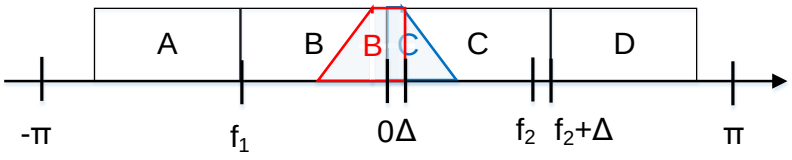


Figura 2.8: Espectro da frequência do sinal reconstruido

É necessário então, proceder à sincronização dos dois receptores. Na figura 2.9 é apresentado o modelo de sincronização digital. Este modelo consiste num correlador que irá comparar

o espectro dos dois canais até conseguir determinar qual o deslocamento entre eles que irá ser uma estimativa Δ' do factor $-\Delta$ apresentado na figura 2.5 sendo de seguida efectuada a devida compensação que alinhará os dois canais permitindo então a sua correcta adição.

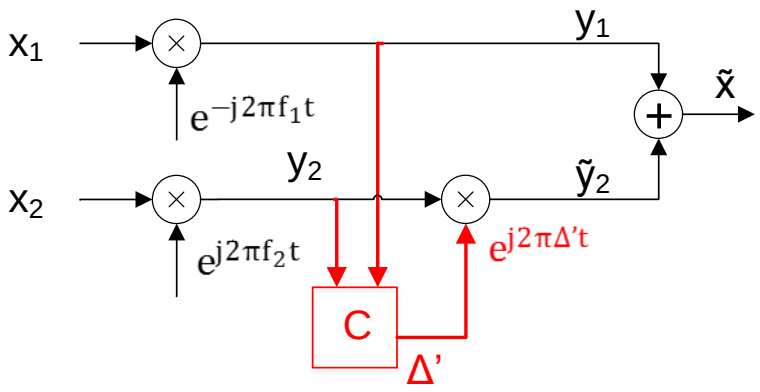


Figura 2.9: Diagrama de blocos da operação de síntese com sincronização

O correlador irá realizar a correlação dos dois sinais na frequência usando a equação 2.21 em que Y_1^* representa o conjugado da resposta em frequência do sinal y_1 e Y_2 a resposta em frequência do sinal y_2:

$$R_{Y_1Y_2} \Rightarrow (Y_1 \star Y_2)[f] = \sum_{m=-\infty}^{\infty} (Y_1^*(m)Y_2(m+f)) \tag{2.21}$$

A estimativa de Δ é obtida através do calculo do máximo do módulo da correlação:

$$\Delta' = max(|R_{Y_1Y_2}|) \tag{2.22}$$

O sinal $\tilde{y}_2$ resulta da compensação efectuada e pode ser adicionado correctamente com o sinal y_1 dando origem ao sinal $\tilde{x}$, uma reconstrução do sinal x.

$$\begin{aligned}\tilde{y}_2 =& y_2 \times e^{j\Delta' t} \\ =& \frac{1}{2}\Big[A_A(e^{j2\pi(F_A+\Delta+\Delta')t} + e^{-j2\pi(F_A-\Delta-\Delta')t}) + A_B(e^{j2\pi(F_B+\Delta+\Delta')t} + e^{-j2\pi(F_B-\Delta-\Delta')t}) + \\ & A_C(e^{j2\pi(F_C+\Delta+\Delta')t} + e^{-j2\pi(F_C-\Delta-\Delta')t}) + A_D(e^{j2\pi(F_D+\Delta+\Delta')t} + e^{-j2\pi(F_D-\Delta-\Delta')t})\Big]\end{aligned} \tag{2.23}$$

Como $\Delta + \Delta' \approx 0$ o sinal reconstruído $\tilde{x}$ fica na forma:

$$\tilde{x}(t) = \frac{1}{2}\Big(A_A[e^{j2\pi F_A t} + e^{-j2\pi F_A t}] + A_B[e^{j2\pi F_B t} + e^{-j2\pi F_B t}] + A_C[e^{j2\pi F_C t} + e^{-j2\pi F_C t}] + A_D[e^{j2\pi F_D t} + e^{-j2\pi F_D t}]\Big) \tag{2.24}$$

Apesar de não ser vísivel no domínio de frequência, os sinais de síntese y_1 e y_2 também irão ter erros introduzidos ao nível da fase. De uma forma análoga à anterior é realizada uma correlação, mas neste caso, ao longo do tempo para determinar a diferença de fase introduzida pela heteródinagem. A correlação entre os sinais calcula-se usando a expressão:

$$R_{y_1 y_2} \Rightarrow (y_1 \star y_2)[n] = \sum_{m=-\infty}^{\infty} (y_1^*(m) y_2(m+n)) \tag{2.25}$$

Para determinar qual o possível atraso de fase calcula-se o valor máximo do módulo da correlação $R_{y_1 y_2}$:

$$\Phi' = max[real(R_{y_1 y_2})] \tag{2.26}$$

2.3 Compensação do Desvio de Frequência e de Fase

De modo a testar o funcionamento do algorimto apresentado na secção anterior foi gerado no Matlab um sinal que possui 8 componentes de diferentes frequências (-17.5KHz, -12.5KHz, -7.5KHz, -2.5KHz, 2.5KHz, 7.5KHz, 12.5KHz e 17.5KHz). Este sinal já se encontra convertido em banda base facto que permite considerar todos os componentes como sendo digitais.

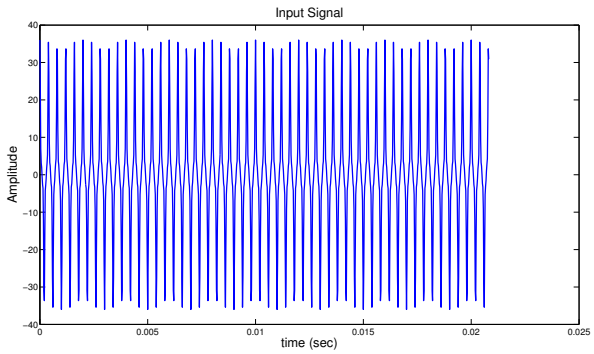


Figura 2.10: Sinal de entrada no tempo

Através do cálculo da FFT do sinal $x(t)$ é possível observar a sua resposta em frequência:

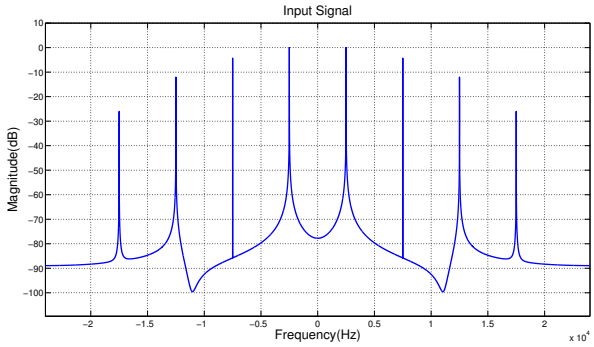


Figura 2.11: Sinal de entrada na frequência

O sinal de entrada é então modelado pelas respectivas frequências centrais f_1 e f_2 (figura 2.12) e de seguida filtrado pelos filtros passa-baixo (figura 2.13), usando o bloco de análise representado na figura 2.5. Refira-se que o sinal X_2 já apresenta um erro de $\Delta = 1000$ introduzido pelo oscilador local.

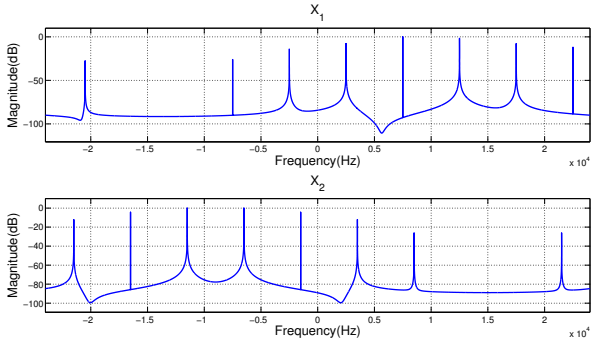


Figura 2.12: Sinal X_1 e X_2

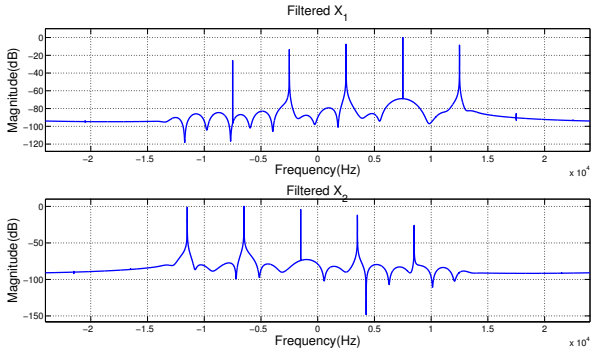


Figura 2.13: Sinal X_1 e X_2 filtrados

De modo a reconstruir o sinal de entrada é necessário voltar a deslocar os espectros para poderem ser somados é necessário proceder á síntese dos mesmo, através do bloco de síntese da figura 2.7. Os sinais resultantes da síntese y_1 e y_2 podem ser observados no domínio da frequência na figura 2.14:

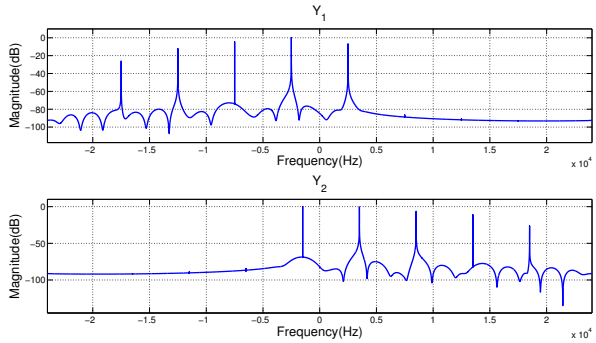


Figura 2.14: Sinais filtrados

Devido ao erro de Δ que o sinal Y_2 contém, o sinal reconstruído irá apresentar componentes de distorção (figura 2.15), componentes estas que alteram considerávelmente o sinal resultante no domínio do tempo.

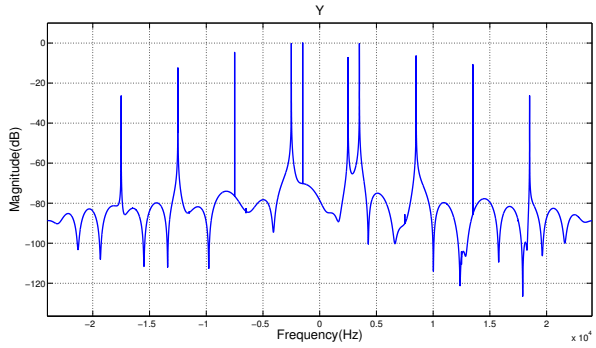


Figura 2.15: Sinal reconstruído com erro representado na frequência

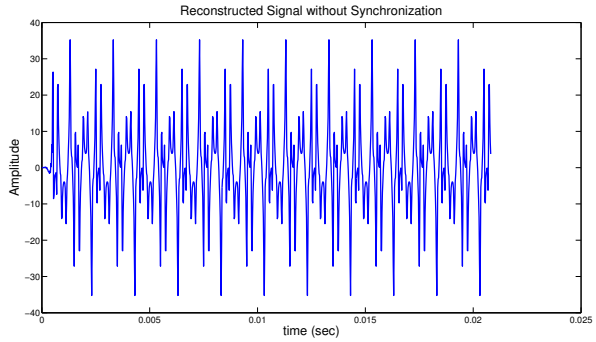


Figura 2.16: Sinal reconstruído com erro representado no tempo

Procede-se então à correlação na frequência para determinar qual foi o desvio introduzido pelo oscilador local. O erro de frequência corresponde à frequência onde ocorre o máximo do módulo da correlação na frequência dos dois sinais (figura 2.17):

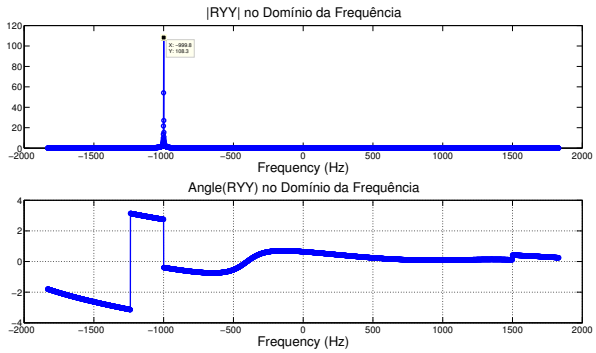


Figura 2.17: Correlação na frequência entre Y_1 e Y_2

Por forma a determinar o máximo do módulo da correlação com uma maior precisão, é necessário efectuar uma interpolação dos pontos envolventes do máximo, para assim, se obter uma maior resolução na zona de interesse.

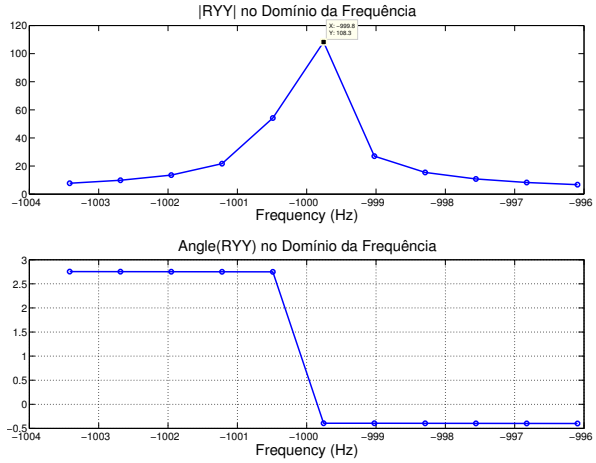


Figura 2.18: Zoom da correlação na frequência entre Y_1 e Y_2

O mesmo processo é realizado no tempo para determinar o desvio de fase Φ. Este valor é apresentado em $\circ$ sendo por isso necessário efectuar a sua conversão para radianos antes da compensação

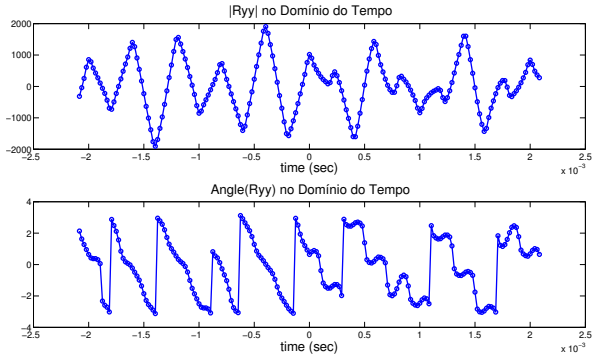


Figura 2.19: Correlação no tempo entre y_1 e y_2

Aplicando a devida compensação é então possível reconstruir uma aproximação do sinal original como é possível verificar nas figuras 2.20 e 2.21, correspondentes ao sinal reconstruído representado no domínio da frequência e do tempo, respectivamente:

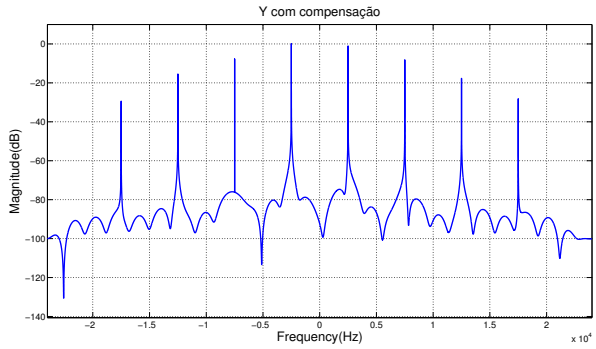


Figura 2.20: Resposta em frequência do sinal reconstruído com compensação

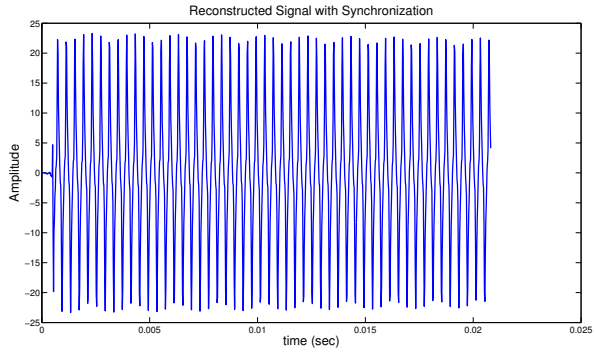


Figura 2.21: Sinal reconstruído com compensação de frequência

Após a realização desta simulação é possível comprovar que o algoritmo desenvolvido consegue determinar e compensar os erros de frequência e fase que são introduzidos de uma forma controlada. Na secção seguinte, o algoritmo será aplicado a sinais gerados por geradores de RF não se sabendo à priori quais os erros introduzidos pelos receptores heteródinos.

Capítulo 3

Hybrid Filters Bank

3.1 Introdução

Mitola propôs o conceito de Software Defined Radio (SDR) [26] ideal como sendo um sistema que realiza todas as operações de processamento no domínio digital. Este sistema consistiria numa antena que capta o sinal Radio Frequency (RF) e por uma Analog-Digital Converter (ADC) que converte o sinal Radio Frequency num sinal digital para ser processado por um Digital Signal Processor (DSP).

Posteriormente surgiu o conceito de Cognitive-Radio (CR) também proposto por Mitola [28] que consiste num rádio que tem a capacidade de se configurar perante a função a realizar desejada. Este conceito abriu caminho a vários estudos, uma vez que apresenta variadíssimas aplicações devido à sua versatilidade.

Uma das abordagens feitas a este conceito designa-se de *Cochlear Radio* [1] [2] [7]. Esta abordagem consiste numa analogia entre o comportamento de um banco de filtros e a cóclea humana. Na figura 3.1 pode-se observar a constituição da cóclea humana e na figura 3.2 é possível verificar o seu comportamento em relação às frequências da gama audível [20Hz:20kHz] me que cada secção da cóclea comporta-se como um filtro passa-banda:

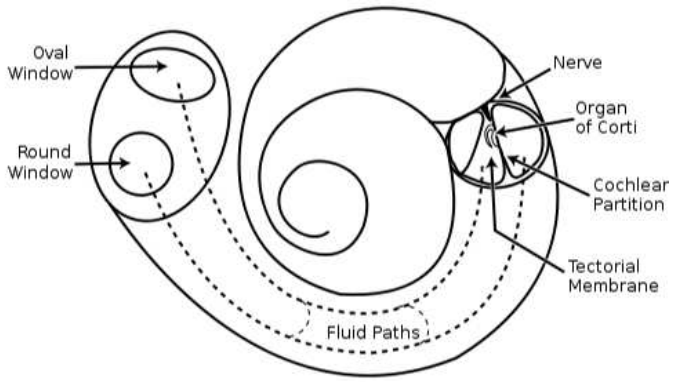


Figura 3.1: Diagrama estrutural da cóclea.

http://en.wikipedia.org/wiki/Cochlea

Dependendo da frequência recebida diferentes zonas da cóclea vão ser excitadas, despoletando vibrações nas zonas ao longo da membrana basilar. Dentro da cóclea também se encontram as chamdas células ciliadas (do inglês *hair cells*) que captam as vibrações na membrana basilar e as convertem em pulso eléctricos, sendo assim consideradas como um conjunto de ADCs de baixa resolução do aparelho auditivo.

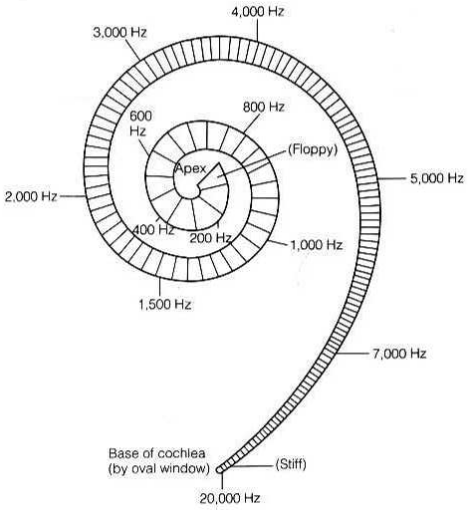


Figura 3.2: Frequências da cóclea.

http://www.ai.rug.nl/acg/cpsp/docs/cochleaModel.html

Através destas duas comparações podemos desenhar um Cognitive-Radio baseado na cóclea humana em que o diagrama de blocos é apresentado na figura 3.3:

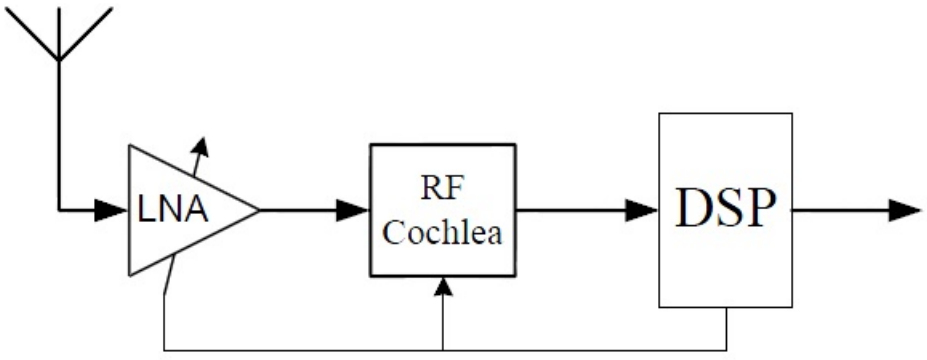


Figura 3.3: Diagrama de blocos do *Cochlear Radio*.

Numa primeira análise, este conceito apresenta um entrave na sua implementação devido

ao facto de ainda não existirem ADCs suficientemente poderosas que permitam captar sinais de grande largura de banda. Uma maneira de ultrapassar esta barreira é usar o conceito de Banco de Filtros [3], em que o sinal RF é dividido em vários sinais com largura de banda M vezes mais pequena antes de serem convertidos digitalmente, sendo M o número de canais.

Este método permite que uma ADCs de largura bastante elevada, na ordem das centenas de MHz (ADCs estas, que apresentam elevada complexidade de implementação e exibem elevados consumos energéticos), como é necessário no projecto de um SDR, seja implementada através de M ADCs que operem a uma frequência M vezes menor permitindo obter assim, uma diminuição no consumo do bloco de digitalização.

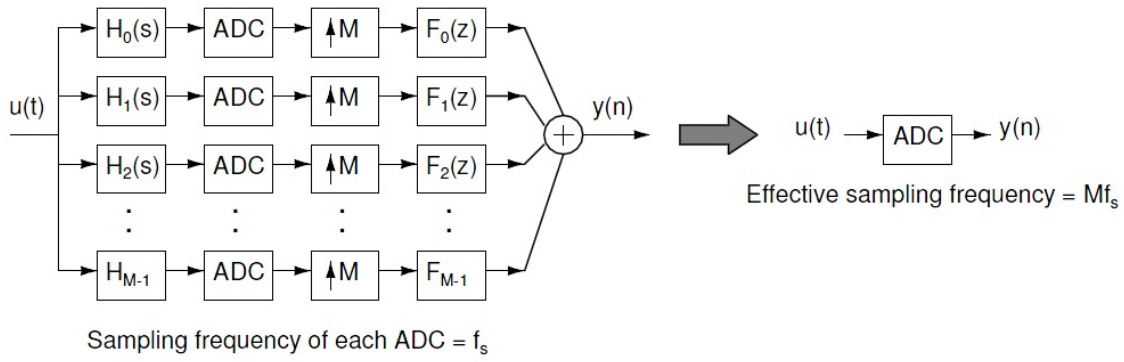


Figura 3.4: Estrutura generalizada de um Hybrid Filter Bank

Este método permite-nos usar ADCs M vezes mais lentas do que se a implementação fosse feita apenas com uma (figura 3.4) e é designado por banco de filtros híbrido porque são utilizados filtros analógicos para análise e filtros digitais para síntese.

3.2 Teoria

3.2.1 Caso Geral

Como referido anteriormente, o banco de filtros permite separar o sinal RF em M sinais distintos com largura M vezes mais baixa. O banco de filtros é constituído por M filtros analógicos e M ADCs, que constituem o bloco de análise, sendo à posteriori calculados os M filtros de síntese digitais que em conjunto com os M interpoladores formam o bloco de síntese. Os filtros de síntese têm por objectivo inverter os filtros de análise assim como compensar as não-linearidades dos componentes analógicos.

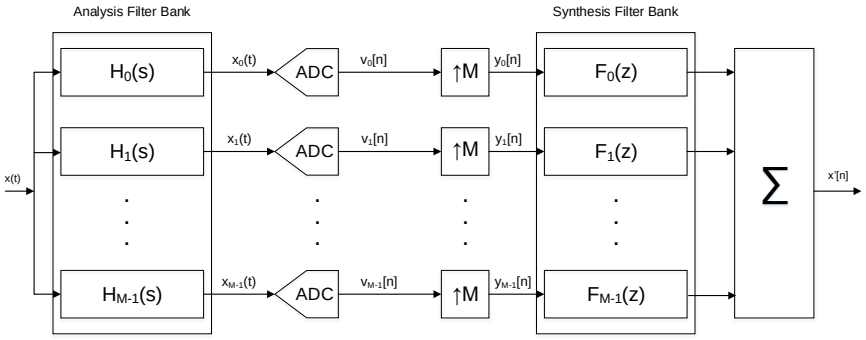


Figura 3.5: Hybrid Filter Bank de M-Canais baseado em ADCs

O banco de filtros analógicos $H_m(j\omega)$ separa o sinal $x(t)$ em M bandas que origina os sinais $X_m = X(jw)H_m(j\omega)$, em que $X(j\omega)$ é a transformada de Fourier do sinal de entrada $x(t)$ com ω em rad/s. Considerando que as ADCs são ideais cada uma destas bandas é digitalizada a uma frequência de $1/MTs$. Perante estes factos é possível calcular a transformada de Fourier do sinal de saída $x\prime[n]$ como:

$$X'(e^{j\omega}) = \frac{1}{MT_s} \sum_{p=-\infty}^{+\infty} X\left(j\frac{\omega}{T_s} - j\frac{2\pi p}{MT_s}\right) \sum_{m=0}^{M-1} H_m\left(j\frac{\omega}{T_s} - j\frac{2\pi p}{MT_s}\right) F_m(e^{j\omega}) \tag{3.1}$$

De forma a simplificar o tratamento matemático considera-se que o sinal $x(t)$ é limitado em frequência permitindo representar o banco de filtros apenas com elementos digitais sendo $\tilde{X}(e^{j\omega})$ e $\tilde{H}_m(e^{j\omega})$ as aproximações digitais de $X(e^{j\omega})$ e $H(e^{j\omega})$, respectivamente com um período de $(2\pi/T_s)$.

Perante esta consideração é possível representar os filtros analógicos de análise por filtros digitais resultando o diagrama da figura 3.6.

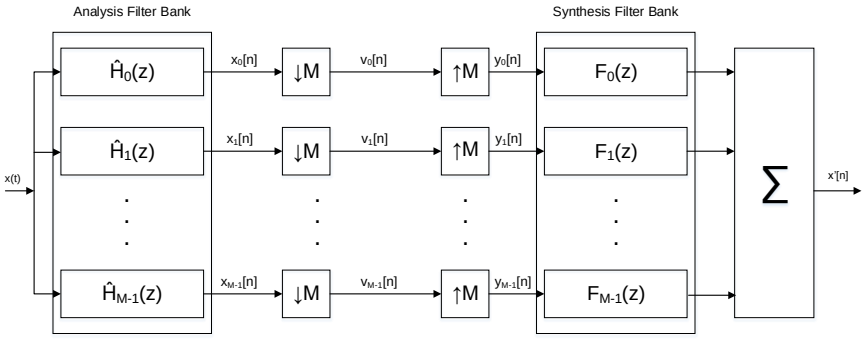


Figura 3.6: Banco de filtros digital equivalente

Desta aproximação a equação 3.1 pode ser escrita na forma:

$$X'(e^{j\omega}) = \frac{1}{MT_s}\sum_{p=0}^{M-1}\tilde{X}\left(e^{j\left(\omega-\frac{2\pi}{M}p\right)}\right)\sum_{m=0}^{M-1}\hat{H}_m\left(e^{j\left(\omega-\frac{2\pi}{M}p\right)}\right)F_m(e^{j\omega}) \tag{3.2}$$

Considerando que:

$$T_p(e^{j\omega}) = \frac{1}{MT_s}\sum_{m=0}^{M-1}\hat{H}_m\left(e^{j\left(\omega-\frac{2\pi}{M}p\right)}\right)F_m(e^{j\omega}) \tag{3.3}$$

A equação 3.2 pode-se reescrever na forma:

$$X'(e^{j\omega}) = \sum_{p=0}^{M-1}X\left(e^{j\left(\omega-\frac{2\pi}{M}p\right)}\right)T_p(e^{j\omega}) \tag{3.4}$$

De modo a conseguir a reconstrução perfeita, o sinal de saída deste sistema será igual ao sinal de entrada com um atraso. Este objectivo é obtido através da imposição da equação 3.5:

$$T_p(e^{j\omega}) = \begin{cases} e^{-j\omega d} & \text{, p=0} \\ 0 & \text{, p=1,...,M-1} \end{cases} \tag{3.5}$$

Em que d representa o atraso do sistema desejado. Da equação 3.5 retiramos que $T_0(e^{j\omega})$ representa a distorção provocado pelo sistema enquanto que $T_p(e^{j\omega})$, com $p = 1, ..., M-1$, representa as diferentes componentes de *aliasing*.

A equação 3.5 pode ser escrita numa forma matricial ficando:

$$\mathbf{T}(e^{j\omega_q}) = \frac{1}{MT_s}\mathbf{H}(e^{j\omega_q})\mathbf{F}(e^{j\omega_q}) \tag{3.6}$$

Em que cada ω_q, em que $q = 0, ..., Q-1$, representa uma frequência usada para a análise em frequência dos filtros de análise. A matriz de análise terá dimensões $M \times M$, sendo M o número de canais do banco de filtros. A partir da equação 3.6 podemos concluir que é necessário resolver Q sistemas de equações independentes, de onde se obtém Q amostras (um para cada ω_q) da resposta em frequência dos filtros de síntese.

De modo a calcular a resposta em frequência é necessário resolver a equação 3.6 em ordem a $F(e^{j\omega_q})$. Sendo a matriz $H(e^{j\omega_q})$ uma matriz quadrada e não-singular é possível obter a sua inversa e assim obter-se a resposta em frequência dos filtros de síntese para cada ω_q usando a equação 3.7:

$$\mathbf{F}(e^{j\omega_q}) = MT_s\mathbf{H}^{-1}(e^{j\omega_q})\mathbf{T}(e^{j\omega_q}) \tag{3.7}$$

3.2.2 Caso de 2 canais

Especificando-se para o caso de $M = 2$, através do uso do sistema de equações 3.6 obtém-se o sistema de equações (eq 3.8), em que $\tilde{H}_0$ é a aproximação digital de um Filtro Passa-Baixo (LPF) e $\tilde{H}_1$ a aproximação digital de um Filtro Passa-Alto (HPF):

$$\begin{bmatrix} e^{-j\omega_q d} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{H}_0(e^{j\omega_q}) & \tilde{H}_1(e^{j\omega_q}) \\ \tilde{H}_0(e^{j(\omega_q-\pi)}) & \tilde{H}_1(e^{j(\omega_q-\pi)}) \end{bmatrix}\begin{bmatrix} F_0(e^{j\omega_q}) \\ F_1(e^{j\omega_q}) \end{bmatrix} \tag{3.8}$$

Garantindo que a matriz $H(e^{j\omega_q})$ é uma matriz invertível é possível calcular a resposta em frequência dos filtros de síntese $F(e^{j\omega_q})$ usando a equação 3.9:

$$\begin{bmatrix} F_0(e^{j\omega_q}) \\ F_1(e^{j\omega_q}) \end{bmatrix} = \begin{bmatrix} \tilde{H}_0(e^{j\omega_q}) & \tilde{H}_1(e^{j\omega_q}) \\ \tilde{H}_0(e^{j(\omega_q-\pi)}) & \tilde{H}_1(e^{j(\omega_q-\pi)}) \end{bmatrix}^{-1} \begin{bmatrix} e^{-j\omega_q d} \\ 0 \end{bmatrix} \tag{3.9}$$

Após o calculo da reposta em frequência dos filtros de síntese é possível calcular a resposta impulsional dos mesmos usando a equação 4.9. De uma forma análoga ao caso da análise em frequência será necessário resolver M sistemas de equações independentes, um sistema para cada filtro de síntese.

$$\mathbf{F}_m = \mathbf{W}^H \mathbf{f}_m \tag{3.10}$$

Sendo a matriz $\mathbf{W}$ a matriz da Discrete Fourier Transform (DFT) enquanto que a operação $(.)^H$ representa o conjugado transposto. É pretinente referir que na maior parte das vezes o número de pontos de análise da frequência Q é muito maior que o número de coeficientes dos filtros de síntese o que resulta num sistema de equações sobre-determinado em que é possível calcular a solução óptima através do métodos dos mínimos quadrados perfeitos transformando a equação 4.9 na seguinte forma:

$$\mathbf{f}_m = (\mathbf{W}\mathbf{W}^H)^{-1}\mathbf{W}\mathbf{F}_m = \mathbf{W}^+\mathbf{F}_m \tag{3.11}$$

Onde $\mathbf{W}^+$ é a matriz pseudoinversa de $\mathbf{W}$.

3.3 Simulação do caso de 2 canais

Para a simulação em matlab destes conceitos consideramos o caso mais simples em que o banco de filtros de análise é composto por um $\tilde{H}_0$ (LPF) e um $\tilde{H}_1$ (HPF) (figura 3.7). Este filtros são do tipo Butterworth com ordem $N = 9$ e a frequência de corte $f_c = 0.5$.

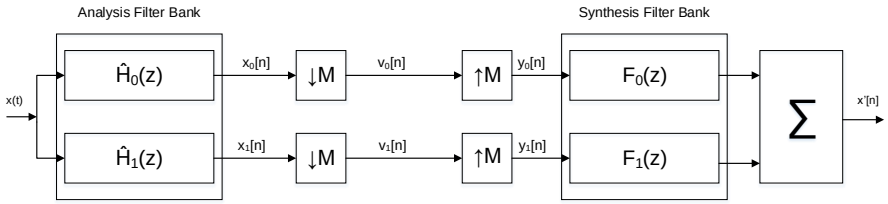


Figura 3.7: Hybrid Filter Bank de M=2 canais

Em seguida foi criado um vector que contem um impulso introduzido às entradas dos filtros para calcular as suas respostas impulsionais que se podem verificar na figura 3.8:

```
% Creation of the analysis filters

  [B1,A1]=butter(No,Fc);          % LowPass Butterworth filter
  [B2,A2]=butter(No,Fc,'high');   % HighPass Butterworth filter
```

```
% Impulse response

  x=[1 zeros(1,Q-1)];              % impulse vector

  h0=filter(B1,A1,x);
  h1=filter(B2,A2,x);
```

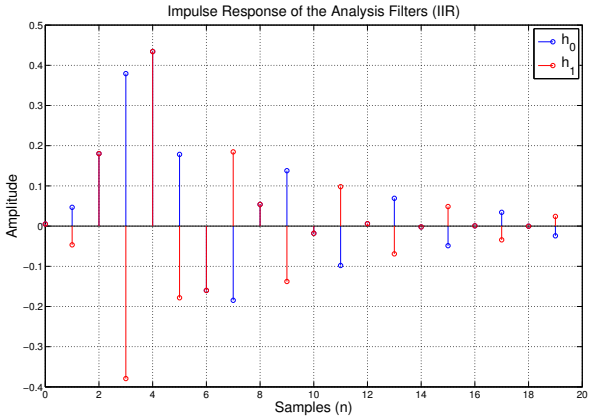


Figura 3.8: Resposta impulsional dos filtros IIR de análise

Calculando a Fast Fourier Transform (FFT) das respostas impulsionais determina-se a resposta em frequência:

$$H(e^{j\omega}) = \sum_{n=0}^{+\infty} h(n)e^{-j\omega} \tag{3.12}$$

```
% Frequency Response
  H0=abs(fft(h0,Q));          % frequency response of LowPass Filter
  H1=abs(fft(h1,Q));          % frequency response of HighPass Filter
```

Na figura 3.9 pode-se observar a resposta em frequência dos filtros de análise:

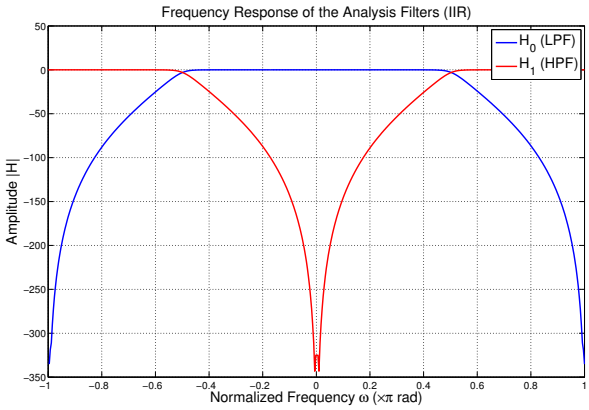


Figura 3.9: Resposta em frequência dos filtros IIR de análise

---

Como já foi referido anteriormente, para o caso $M = 2$ a matriz de análise apresenta a seguinte forma:

$$H(e^{j\omega}) = \begin{bmatrix} \tilde{H}_0(e^{j\omega_q}) & \tilde{H}_1(e^{j\omega_q}) \\ \tilde{H}_0(e^{j(\omega_q-\pi)}) & \tilde{H}_1(e^{j(\omega_q-\pi)}) \end{bmatrix} \tag{3.13}$$

Este cálculo foi efectuado no domínio do tempo através do seguinte código:

```
% Compute Analysis Matrix

  n=1:Q;

  h=zeros(M,M,Q);              % Analysis Matrix
  h(1,1,:)=h0;                 % H_0
  h(1,2,:)=h1;                 % H_1
  h(2,1,:)= (-1).^n.*h0;       % H_0/pi
  h(2,2,:)= (-1).^n.*h1;       % H_1/pi

  H=fft(h,Q,3);
```

Na figura 3.10 é possível observar a resposta em frequência dos 4 filtros:

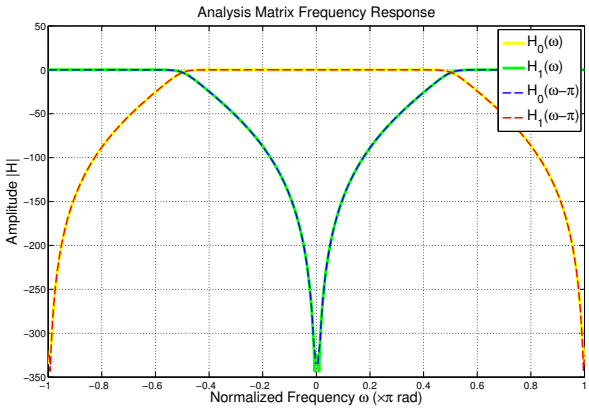


Figura 3.10: FFT da Matriz de análise H

É necessário verificar o número de condição da matriz de análise de modo a confirmar que não existem gamas de frequências com valores muito elevados. Se esse caso se verificar, iremo-nos deparar com um sistema de equações (equação 3.7) mal-condicionado que pode originar erros grandes na solução final.

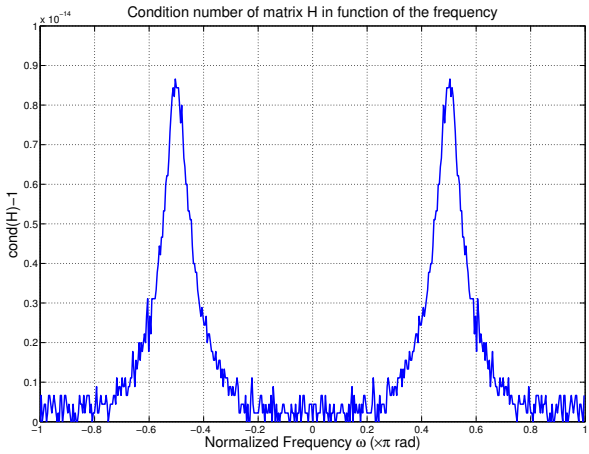


Figura 3.11: Número de condição da matriz de análise H

Pela observação da figura 3.11 confirma-se que os valores do número de condição são baixos ao longo de todas as frequências.

Fazendo uso das equações 3.6 e 3.7 obtém-se a resposta em frequência dos filtros de síntese que se observam na figura 3.12:

```
% Delay Vector
    T=zeros(M,1,Q);
    w= (0:Q-1)*2*pi/Q;
    T(1,1,:)=exp(-j*w*D);

% Compute the Response Frequency of the Synthesis Filters
```

```
F=zeros(M,1,Q);

for i=1:Q
    F(:,1,i)= H(:,:,i) \ T(:,1,i);
end

F=M*transpose(reshape(F,2,Q));
F0=F(:,1);
F1=F(:,2);
```

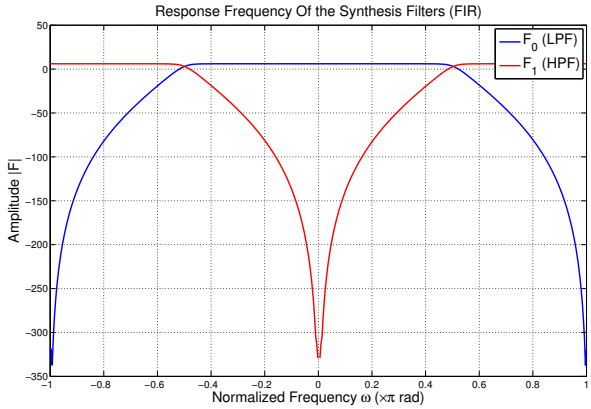


Figura 3.12: Resposta em frequência dos filtros FIR de síntese

Tendo sido calculadas as respostas em frequência dos filtros de síntese pode-se agora usar o método dos mínimos quadrados para calcular a resposta impulsional de cada filtro de síntese usando a equação 3.11 com o L a ser abritrado em 64:

```
% Compute the Impulse Response of the Synthesis Filters

w=dftmtx(Q);                % DFT Matrix
wi=1:Q;
w=w(wi,1:L);                % Length of the Synthesis Filters

f=zeros(L,M);

for i=1:M
    f(:,i)= w \ F(wi,i);    % f_m= w+ F_m
end

f0=real(f(:,1));
f1=real(f(:,2));
```

Na figura 3.13 é possível observar a resposta impulsional dos filtros de síntese onde se verifica que ambas as respostas convergem para zeros em ambos os lados:

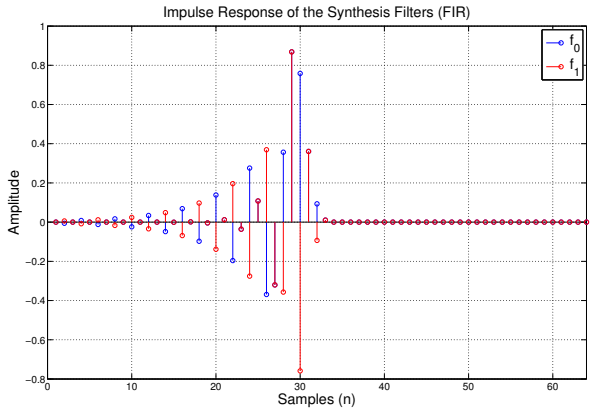


Figura 3.13: Resposta impulsional dos filtros FIR de síntese

De modo a testar o sistema foi criado uma função que simula o banco de filtros de 2 canais (*FilterBankTwoCh*). Esta função tem como entradas o sinal de entrada e as repostas impulsionais dos filtros de análise e dos filtros de síntese e retorna o sinal de saída, em que no caso de existir reconstrução perfeita será uma versão atrasada do sinal de entrada. Para a realização do teste foram criados 2 impulsos com a diferença de um atraso. Os sinais de saída resultantes permitem-nos calcular a distorção e o *aliasing*. O seguinte código foi usado para a simulação:

```
x1= [1; zeros(Ni-1,1)];          % x1= [1 0 0 0 ...]
x2= [0; 1; zeros(Ni-2,1)];       % x2= [0 1 0 0 ...]

y1= FilterBankTwoCh(x1,h0,h1,f0,f1);
y2= FilterBankTwoCh(x2,h0,h1,f0,f1);
y2= [y2(2:end) 0];               % Align the delayed impulse response

T0= abs(fft(y1+y2,Nfft))/2-1;    % Distortion
T1= abs(fft(y1-y2,Nfft))/2;      % Aliasing
```

Em que os resultados da distorção e do *aliasing* podem ser observados na figura 3.14:

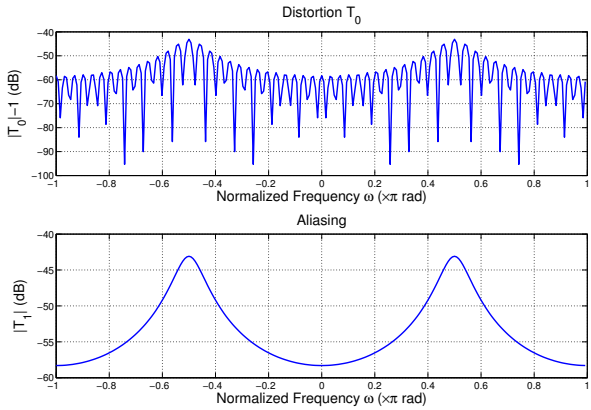
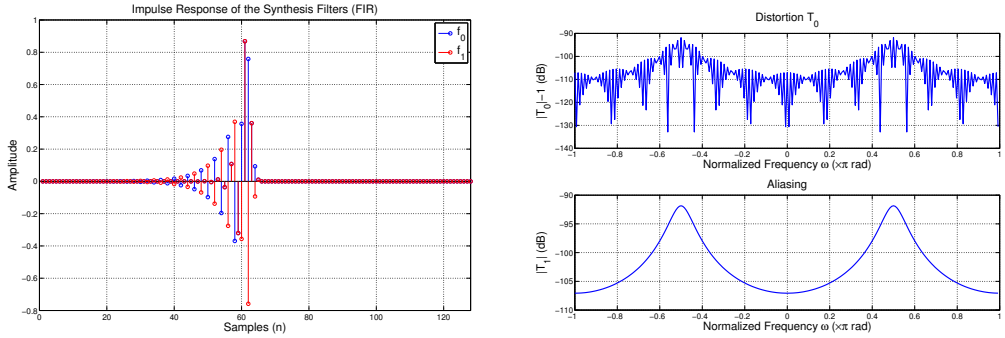


Figura 3.14: Distorção e *Aliasing*

De onde se observa que tanto a distorção como a *aliasing* apresentam valores entre os -50 e os -60 dB com os máximos a ocorrerem na frequência de corte. Significa isto, que ao ser introduzido um impulso na entrada do sistema, o banco de filtros consegue realizar uma reconstrução perfeita com parametros controlados.

3.4 Variação do Comprimento dos Filtros de Síntese

Na necessidade de obter menores valores de distorção e *aliasing* é necessário aumentar o comprimento da resposta impulsional dos filtros de síntese. Para demonstrar este facto foram efectuadas simulações com $L = 128$ (figura 3.15) e $L = 256$ (figura 3.16).

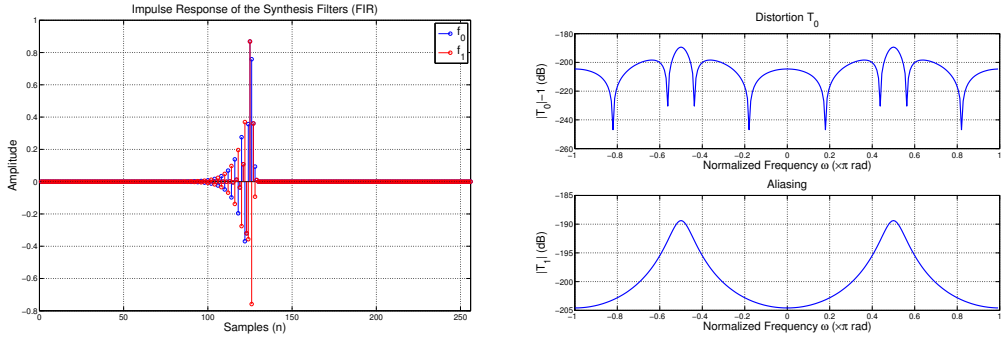


(a) Resposta impulsional

(b) Distorção e *Aliasing*

Figura 3.15: Filtros de síntese FIR com 128 coeficientes

Pela observação da figura 3.15 confirma-se que com a aumento do número de coeficientes para o dobro ($L = 128$) dos filtros de síntese os valores de distorção e *aliasing* descem para valores a rondar os -100 dB.



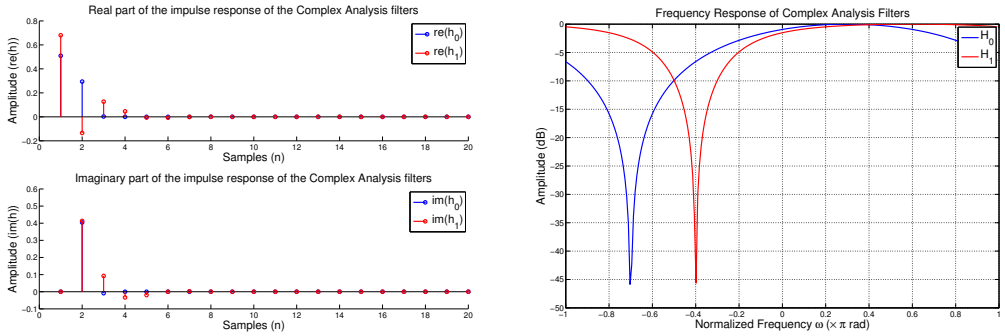
(a) Resposta impulsional (b) Distorção e *Aliasing*

Figura 3.16: Filtros de síntese FIR com 256 coeficientes

No caso de o de $L = 256$ obtém-se ainda menores valores de distorção e *aliasing* que se encontram na casa dos -200 dB. No entanto este aumento introduz também um maior número de cálculos a efectuar. Perante esta situação é desejado obter um acordo entre os valores de distorção e o peso de processamento.

3.5 Filtros Complexos em Hybrid Filter Bank

Como será abordado no capítulo 5, o sistema em estudo é constituido, não por filtros reais mas, por complexos (implementação do Front-End (FE) será explicada na secção 5.1). Este pormenor obriga a ter em consideração alguns aspectos desprezados nas secções anteriores.



(a) Resposta impulsional (b) Resposta em frequência

Figura 3.17: Resposta impulsional e resposta em frequência dos filtros de análise IIR Complexos

A primeira grande diferença é visível na resposta em frequência dos filtros IIR de análise (figura 3.17(b)) onde se verifica que não existe simetria par ao longo do eixo das frequências ω como acontecia no caso de filtros reais. No caso de sinais complexos, estes apenas apresentam informação relevante ao longo das frequências positivas [31]. Esta caracteristica apresenta vantagem em relação aos filtros reais, uma vez que, quando se efectua *down-conversion* para a banda-base não ocorre sobreposição da componente positiva com a componente negativa.

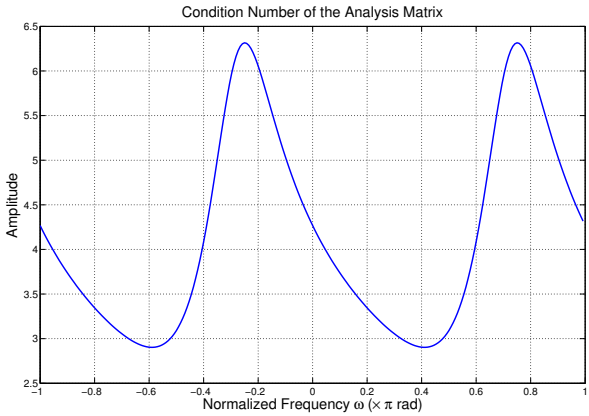
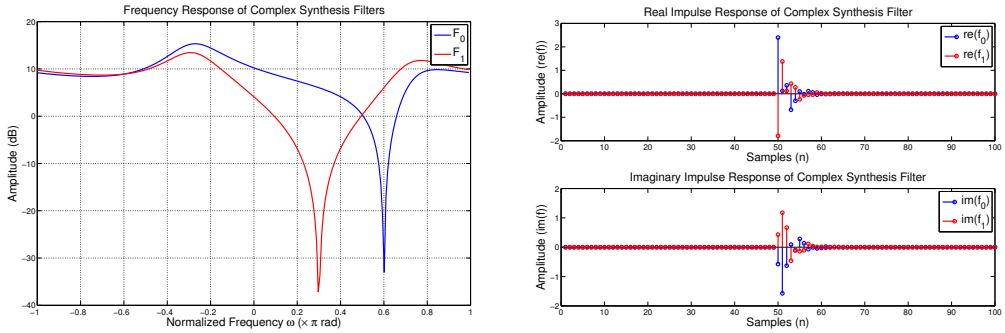


Figura 3.18: Número de condição da matriz de análise complexa

O número de condição da matriz de análise (figura 3.18) apresenta valores baixos, o que representa a garantia de que as soluções obtidas através da equação 3.6 para os filtros de síntese apresentarão erros baixos, devido ao facto de o sistema de equações ser bem condicionado.

Após a verficação da condição do sistema de equações, é possível proceder ao cálculo dos filtros de síntese usando a equação 3.7 para a obtenção da resposta em frequência (figura 3.19(a)) e a equação 3.11 para o cálculo da resposta impulsional (figura 3.19(b)).



(a) Resposta em frequência

(b) Resposta impulsional

Figura 3.19: Resposta em frequência e resposta impulsional dos filtros de síntese IIR Complexos

Através do cálculo da distorção e do *aliasing* (figura 3.20) provocados pelo sistema é possível concluir que os filtros de síntese conseguem efectuar uma reconstrução perfeita (valores de distorção na ordem dos -160 dB e *aliasing* a rondar os -190 dB).

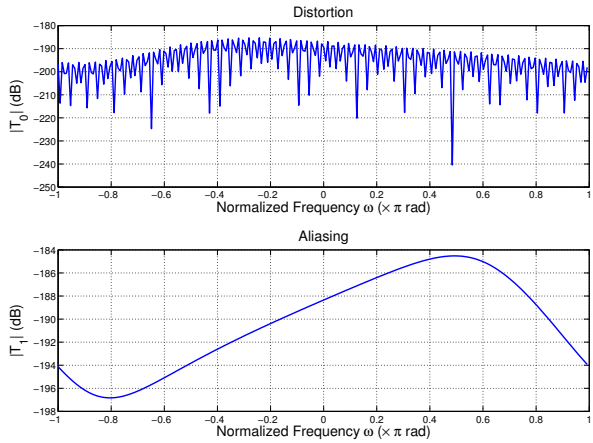


Figura 3.20: Distorção e *Aliasing* do banco de filtros complexos

---

Concluindo-se que, com a utilização de filtros complexos é possível obter uma melhor reconstrução do sinal $x(t)$ em comparação com os filtros reais.

Capítulo 4

Projection on Convex Sets

4.1 Introdução

Neste capítulo será abordada a inversão de filtros através do algoritmo Papoulis-Gerchberg que é um método do tipo Projection On Convexs Sets (POCS). Uma das razões para a utilização deste método prende-se com o facto de na maior parte das situações ser apenas necessário inverter uma zona específica de interesse. Uma vez que, este método apenas efectua cálculos na zona de interesse o sistema de equações 3.6 pode ser diminuído de Q sistemas de equações para Q_n, em que Q_n representa o número de equações referente à banda de interesse, permitindo assim alcançar um menor consumo de energia, como será explicado detalhadamente ao longo deste capítulo.

4.2 Algoritmo Papoulis-Gerchberg

Um dos métodos de adquirir sinais com uma elevada largura de banda é a de separar o sinal de entrada em várias bandas e consequentemente digitalizar as mesmas através de M ADCs com uma frequência de amostragem M vezes mais pequena do que se fosse efectuado apenas com uma. Esta abordagem é conhecida como Hybrid Filter Bank (HFB), que foi estudada no capítulo 3 e cujo diagrama de blocos está representado na figura 4.1:

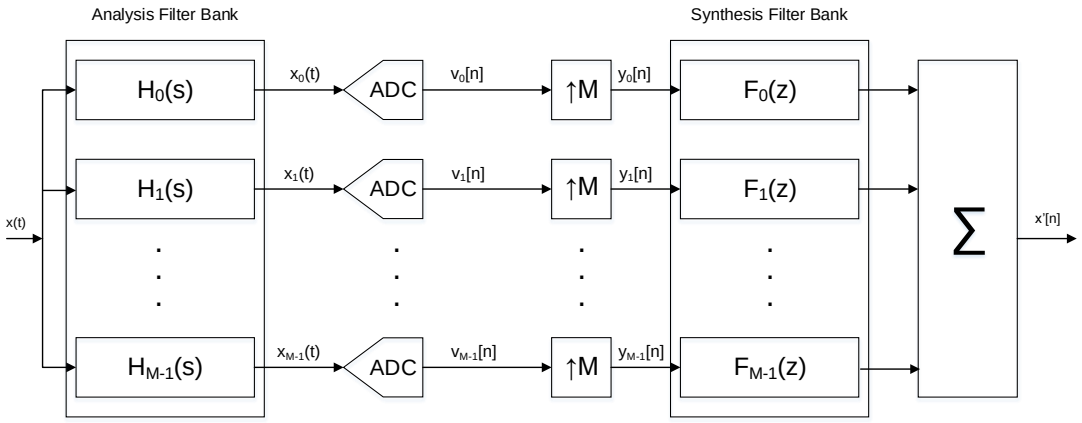


Figura 4.1: Estrutura de HFB de M-Canais

A transformada de Fourier do sinal de saída calcula-se através da equação 4.1:

$$X'(e^{j\omega}) = \frac{1}{MT_s} \sum_{p=-\infty}^{+\infty} X\left(j\frac{\omega}{T_s} - j\frac{2\pi p}{MT_s}\right) \sum_{m=0}^{M-1} H_m\left(j\frac{\omega}{T_s} - j\frac{2\pi p}{MT_s}\right) F_m(e^{j\omega}) \tag{4.1}$$

Assumindo que o sinal de entrada $x(t)$ possui uma banda limitada de frequencias, isto é, que $|X(j\Omega)| = 0$ para $|\Omega| > \frac{\pi}{T_s}$ pode-se considerar o banco de filtros composto apenas por elementos digitais como pode ser observado na figura 4.2:

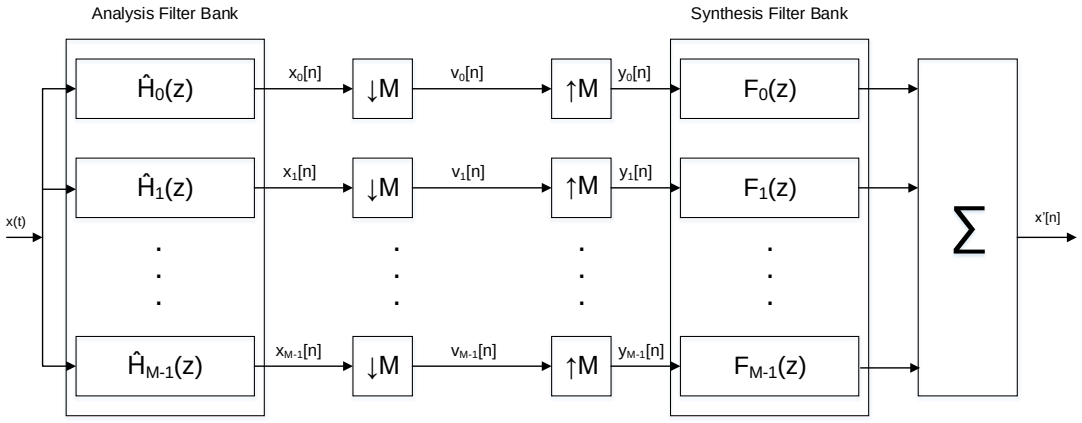


Figura 4.2: Hybrid Filters Bank digital equivalente

Usando esta análise a equação 4.1 fica no forma:

$$X'(e^{j\omega}) = \frac{1}{MT_s} \sum_{p=0}^{M-1} \tilde{X}\left(e^{j\left(\omega - \frac{2\pi}{M}p\right)}\right) \sum_{m=0}^{M-1} \hat{H}_m\left(e^{j\left(\omega - \frac{2\pi}{M}p\right)}\right) F_m(e^{j\omega}) \tag{4.2}$$

É possivel reescrever a equação 4.2 na forma:

$$X'(e^{j\omega}) = \sum_{p=0}^{M-1} X\left(e^{j\left(\omega - \frac{2\pi}{M}p\right)}\right) T_p(e^{j\omega}) \tag{4.3}$$

Considerando que:

$$T_p(e^{j\omega}) = \frac{1}{MT_s} \sum_{m=0}^{M-1} \hat{H}_m\left(e^{j\left(\omega - \frac{2\pi}{M}p\right)}\right) F_m(e^{j\omega}) \tag{4.4}$$

Para obter reconstrução perfeita é necessário impor a condição de o sinal de saída ser igual ao sinal de entrada com um atraso provocado pelos filtros do banco. Esta condição é realizada através da seguinte equação:

$$T_p(e^{j\omega}) = \begin{cases} e^{-j\omega d} & , p = 0 \\ 0 & , p = 1, ..., M-1 \end{cases} \tag{4.5}$$

Onde d é o atraso desejado pelo sistema. Normalmente $T_0(e^{j\omega})$ é designado pelo termo de distorção enquanto que $T_p(e^{j\omega})$, com $p = 1, ..., M-1$ de compoentes de *aliasing*.

Neste caso em concreto o objectivo é inverter filtros do tipo passa-banda usando um método do tipo POCS designado por Papoulis-Gerchberg. Este método é implementado de uma forma iterativa e conduz a uma redução de consumo energético uma vez que só são efectuados cálculos numa determinada banda de frequências de interesse designada por ω_I. As várias iterações executam uma reavaliação da resposta em frequência dos filtros de síntese digitais e a truncagem da reposta impulsional para assegurar baixa distorção e *aliasing* na banda de interesse ignorando o que se passa fora desta.

Em seguida são descritos os passos do algoritmo de Papoulis-Gerchberg:

1. É determinada a reposta em frequência dos filtros de síntese com um certo número de coeficientes L.
2. É definida a região de interesse ω_I e a de não interesse ω_N de modo a que $\omega_I \bigcup \omega_N = [-\pi : \pi]$.
3. É calculada a função transferência do Banco de Filtros através da equação 4.4 resultando $\tilde{T}_p(e^{j\omega})$.
4. Do passo anterior é necessário assegurar que:

$$\tilde{T}_p(e^{j\omega}) = \begin{cases} T_p(e^{j\omega}) & \text{for } \omega = \omega_I \\ \tilde{T}_p(e^{j\omega}) & \text{for } \omega = \omega_N \end{cases} \tag{4.6}$$

5. É calculada novamente a resposta em frequência dos filtros de sintese usando $\tilde{T}_p(e^{j\omega})$ como objectivo a ser atingido.

6. É efectuada a truncagem da resposta impulsional dos filtros de síntese $f_m(n)$ através da equação 4.7

$$\tilde{f}(n) = f_m(n)W(n), m = 0, 1, ..., M-1 \tag{4.7}$$

Em que $W(n)$ é uma "box function"definada por:

$$W(n) = \begin{cases} 1 & \text{para } 0 < n < d - |L'/2| \text{ e } d + |L'/2| < n < L' \\ 0 & \text{outros casos} \end{cases} \tag{4.8}$$

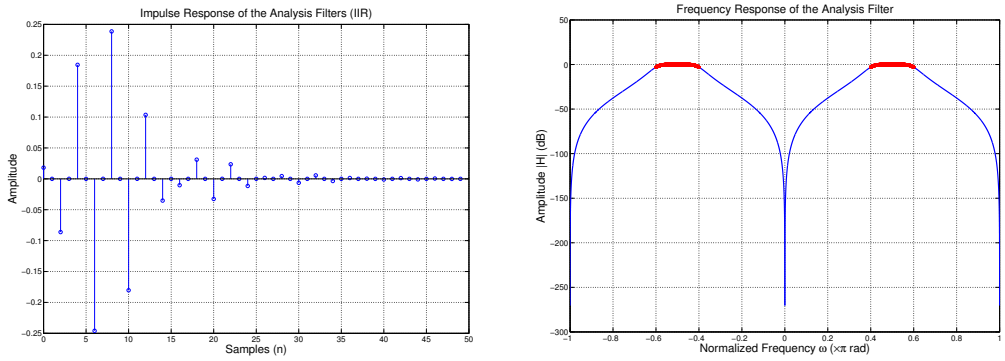
Onde d representa o atraso da resposta do banco de filtros.

7. Chegado a este ponto entra-se num processo iterativo voltando ao ponto 3 usando $\tilde{f}_m$ para calcular $F_m(e^{j\omega})$.

4.3 Aplicação do algoritmo Papoulis-Gerchberg no banco de filtros

4.3.1 Realização do algoritmo Papoulis-Berchberg com 1 filtro

Numa primeira abordagem à realização deste método foi considerado o caso mais simples de apenas um Filtro Passa-Banda (BPF) do tipo *Butterworth* de ordem $N = 3$. Foi escolhido este tipo de filtros porque são os que apresentam uma banda de passagem mais *flat*. A resposta impulsional e em frequência podem ser observadas nas figuras 4.3(a) e 4.3(b) respectivamente.



(a) Resposta impulsional (b) Resposta em frequência

Figura 4.3: Resposta impulsional e resposta em frequência do Filtro Passa-Banda de Análise.

Para a implementação da reconstrução perfeita foi imposto um atraso de $L/2$, usando um vector da forma $T_0 = e^{-j\omega\frac{L}{2}}$.

A inversão do filtro, de modo a obter reconstrução perfeita, é feita somente dentro da zona de interesse definida através da visualização da resposta em frequência do filtro e foi definido um *threshold* de -3 dB para a mesma (assinalado a vermelho).

De modo a impor que a resposta a impulso seja real é necessário forçar a que as componentes negativas da resposta em frequência sejam o conjugado das componentes positivas, o que

origina uma simetria par na aquisição das compoentes negativas. Essa simetria é observada na figura 4.4.

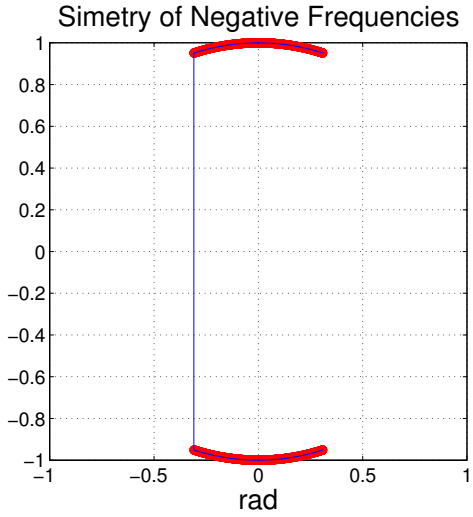
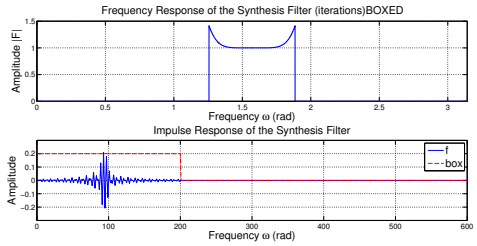
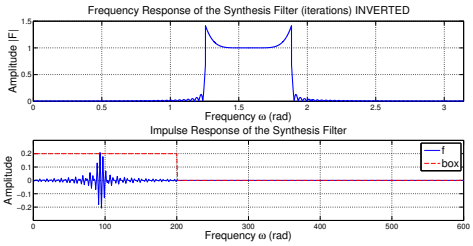


Figura 4.4: Simetria das frequências na zona de interesse ω_I

A partir desta altura pode-se aplicar o algoritmo de Papoulis-Gerchberg analisado na secção anterior. Numa primeira simulação foram arbitradas 100 iterações com os filtros de síntese a possuirem 200 coeficientes. Nas figuras 4.5(a) e 4.5(b) é possível observar os passos descritos na secção anterior.



(a) Primeira iteração do algoritmo - inversão no domínio da frequência

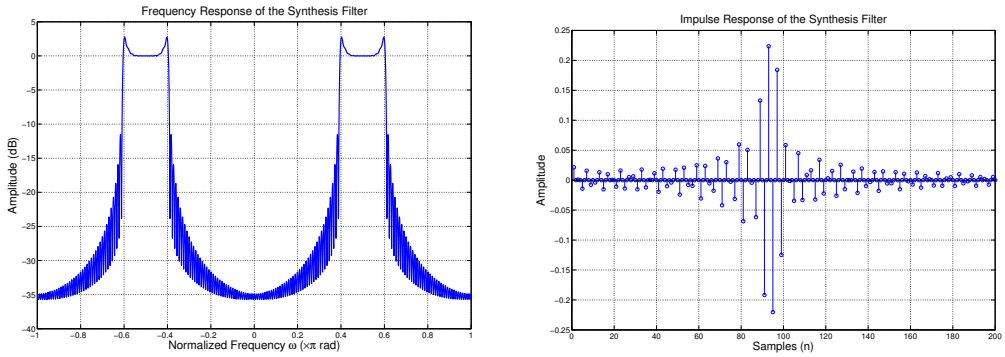


(b) Primeira iteração do algoritmo - truncagem no domínio do tempo

Figura 4.5: Primeira iteração do algortimo

É possivel verificar que a imposição de inversão no domínio da frequência origina alterações na resposta impulsional e que por sua vez, a truncagem no domínio do tempo origina lobulos secundários na resposta em frequência devido ao facto de estarmos a multiplicar a resposta impulsional com um sinal rectangular que possui uma resposta em frequência da forma de uma $sinc(x)$, em que $sinc(x) = sin(x)/x$.

Após realizadas as 100 iterações obtém-se a resposta em frequência e impulsional do filtro inverso (figuras 4.6(a) e 4.6(b) respectivamente)



(a) Resposta em frequência (b) Resposta impulsional

Figura 4.6: Resposta em frequência e resposta impulsional do filtro de síntese

---

De modo a concluirmos que a inversão desejada é implementada com sucesso é calculado o produto da resposta em frequência do filtro inverso com a resposta em frequência do filtro de análise, que teóricamente seria igual a 1 em toda a gama de frequência, mas que no nosso caso só o é na gama de frequências de interesse.

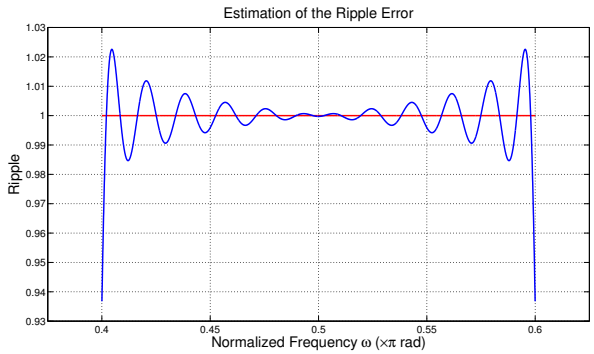
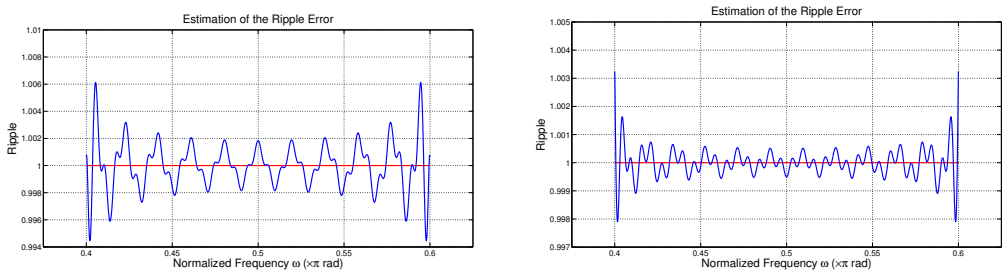


Figura 4.7: *Ripple* na banda de interesse ω_I após 100 iterações

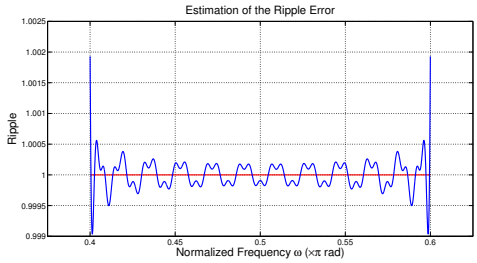
Para diminuir o ripple de erro visível na figura 4.7 é necessário um maior número de iterações, ou por outro lado, considerar filtros de síntese com um maior número de coeficientes. De seguida foram efectuadas várias simulações com diferentes parametros, número de iterações e comprimento dos filtros de síntese.

Pela comparação das figuras 4.8 e 4.9 conclui-se que também é possível diminuir o *ripple* através do aumento do tamanho dos filtros de síntese.

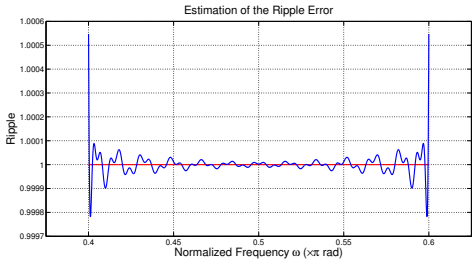


(a) 1000 iterações

(b) 10000 iterações



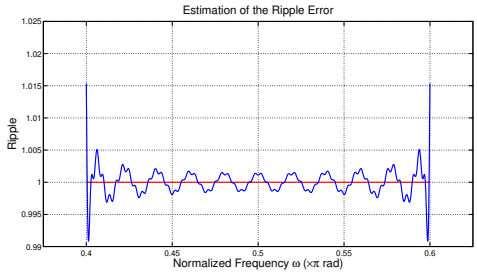
(c) 100000 iterações



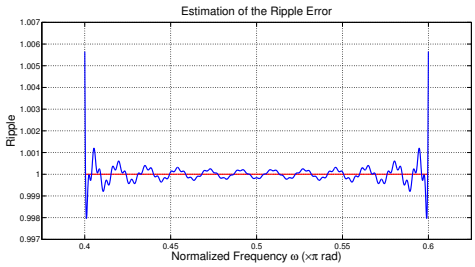
(d) 1000000 iterações

Figura 4.8: Comparação das várias simulações com diferentes valores de iterações com filtros de síntese com $L = 400$ coeficientes

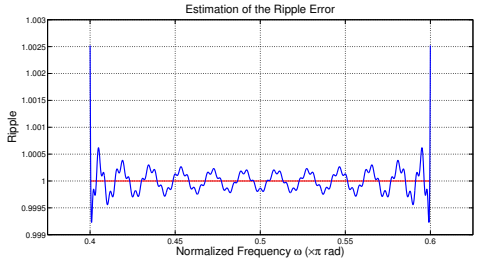
---



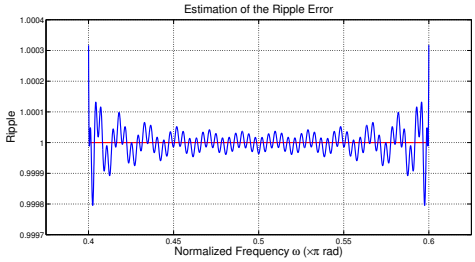
(a) 1000 iterações



(b) 10000 iterações



(c) 100000 iterações



(d) 1000000 iterações

Figura 4.9: Comparação das várias simulações com diferentes valores de iterações com filtros de síntese com $L = 600$ coeficientes

Foram também calculados os tempo de execução do processo iterativo para vários valores de iterações (ver tabela 4.1) de modo a auferir como o aumento das iterações influência no

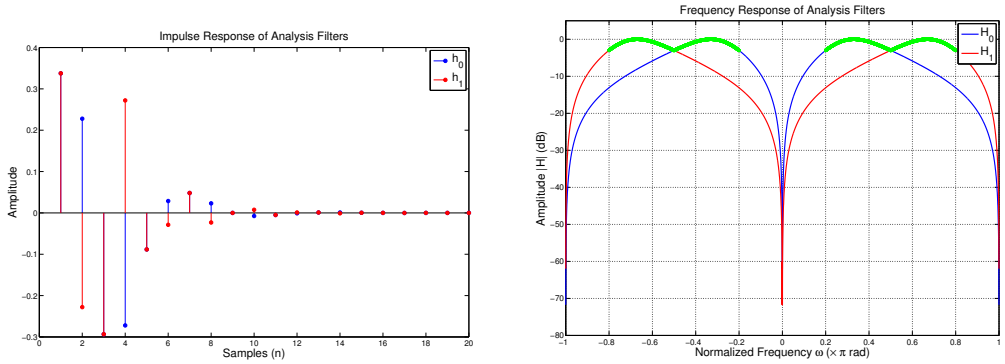
tempo. Estes valores foram registados durante a simulação em ambiente *Matlab* usando um computador dotado de um processador Intel(R) Core i7-4770K @ 3.5 GHz, com 16 GB de memória RAM. É possível concluir que o tempo aumenta de uma forma linear com o aumento do número de iterações.

Niter	time
100	0.103539 segundos
1000	0.999131 segundos
1000	9.691521 segundos
10000	96.931583 segundos

Tabela 4.1: Tempo de simulação do algoritmo Papoulis-Gerchberg aplicado a um filtro com diferentes números de iterações

4.3.2 Realização do algoritmo Papoulis-Berchberg para 2 filtros

Após a conclusão da implementação do algoritmo com um filtro passou-se para o caso de 2 filtros passa-banda (figura 3.7). Neste caso os calculos complicam-se um pouco devido ao facto de ser necessário a inclusão de matrizes. De modo a facilitar a implementação foram usados filtros o mais simples possível. Como estamos a inverter filtos passa-banda a ordem mínima é 2. A resposta a impulso e a resposta em frequência podem ser observadas nas figuras 4.10(a) e 4.10(b), respectivamente:



(a) Resposta impulsional

(b) Resposta em frequência

Figura 4.10: Resposta impulsional e resposta em frequência dos filtros de análise

No caso de um banco de filtros com 2 canais é necessário calcular a matriz de análise na forma

$$\tilde{H}(e^{j\omega_q}) = \begin{bmatrix} \tilde{H}_0(e^{j\omega_q}) & \tilde{H}_1(e^{j\omega_q}) \\ \tilde{H}_0(e^{j(\omega_q-\pi)}) & \tilde{H}_1(e^{j(\omega_q-\pi)}) \end{bmatrix} \tag{4.9}$$

onde para calcular os filtros deslocado de π é necessário proceder a alguns cálculos:

Pensando no caso de possuirmos apenas $N = 8$ amostras na frequência é possível observar esses pontos numa circunferência de raio unitário e fazer os deslocamentos necessários:

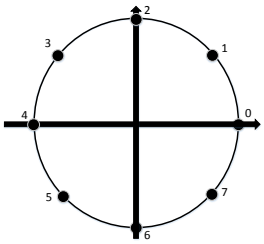


Figura 4.11: Circulo unitário com $N = 8$ pontos na frequência

---

Devido ao facto de o *Matlab* usar índices a começar em 1 é necessário fazer a compensação:

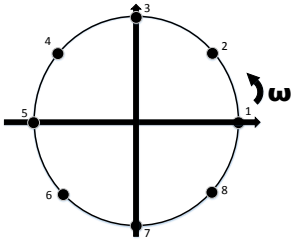


Figura 4.12: Circulo unitário com $N = 8$ pontos na frequência usando índices do *Matlab*

Pela observação da figura 4.12 pode-se verificar a regra matemática para a obtenção da transladação de π:

ω	$\omega - \pi$
1	5
2	6
3	7
4	8
5	1
6	2
7	3
8	4

Tabela 4.2: Correspondência entre ω e $\omega - \pi$

Deste modo é possível verificar que existem duas zonas de conversão:

$k = 1 : N/2$	$\Rightarrow$	$rem(k, N) + N/2$
$k = N/2 + 1 : N$	$\Rightarrow$	$rem(k, n) - N/2$

Tabela 4.3: Relação matemática entre ω e $\omega - \pi$

Onde a operação $rem(k, N)$ representa o resto da divisão inteira de k por N. Foi então implementado o seguinte código *Matlab* para gerar a matriz de análise:

```
for k=1:n
```

```
    if k<=n/2
        H0pi(k)=H0(rem(k,n)+n/2);
        H1pi(k)=H1(rem(k,n)+n/2);
    else
        if k==n
            H0pi(k)=H0(n/2);
            H1pi(k)=H1(n/2);
        else
            H0pi(k)=H0(rem(k,n)-n/2);
            H1pi(k)=H1(rem(k,n)-n/2);
        end
    end
end

H(1,1,:)=H0;
H(1,2,:)=H1;
H(2,1,:)=H0pi;
H(2,2,:)=H1pi;
```

É importante referir que, uma vez que estamos a trabalhar com filtros passa-banda é normal que o determinante da matriz H seja 0 nas frequências $\omega = 0$, $\omega = \pi$ e nas suas respectivas imagens, devido ao facto de um filtro passa-banda ter como objectivo rejeitar todas as componentes de frequência que se encontrem na banda de rejeição, $[0 : \omega_L] \bigcup [\omega_H : \pi]$. Estas frequências correspondem aos picos observados na figura 4.13 que apresenta o número de condição da matriz de análise.

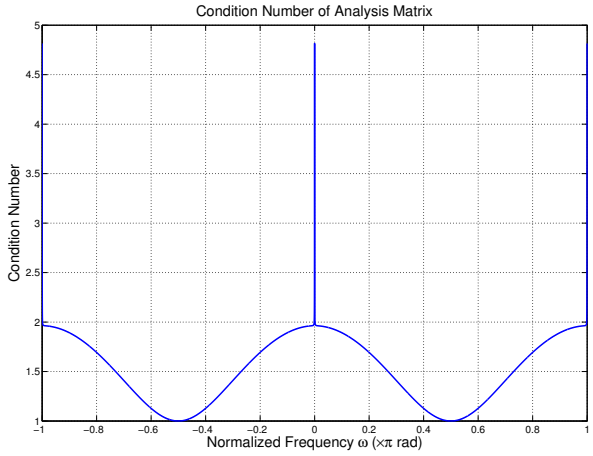


Figura 4.13: Número de condição da matriz de análise

Como o número de condição se mantém consideravelmente baixo pode-se concluir que é possível inverter a matriz de análise obtendo resultados sem obter os sistema de equações 3.6 e 3.7 como mal-condicionados. Os valores nos pontos críticos de índice 1,n/2,n/2+1 e n foram diminuídos através da introdução de um erro na resposta em frequência dos filtros de análise:

```
H0(1)=1e-3*i;
```

```
H1(1)=1e-3*i;
H0(end)=1e-3*i;
H1(end)=1e-3*i;
```

Através deste artefacto é possível evitar a consequência de um determinante da matrix de H ser zero nos mesmos.

```
>> H(:,:,1)

ans =

   1.0e-03 *

   0.0000 + 1.0000i   0.0000 + 1.0000i
   0.0001 + 0.2589i   0.0006 + 0.7968i

>> H(:,:,n/2)

ans =

   1.0e-03 *

   0.0001 - 0.2589i   0.0006 - 0.7968i
   0.0000 + 1.0000i   0.0000 + 1.0000i

>> H(:,:,n/2+1)

ans =

   1.0e-03 *

   0.0001 + 0.2589i   0.0006 + 0.7968i
   0.0000 + 1.0000i   0.0000 + 1.0000i

>> H(:,:,n)

ans =

   1.0e-03 *

   0.0000 + 1.0000i   0.0000 + 1.0000i
   0.0001 - 0.2589i   0.0006 - 0.7968i
```

De referir também que esta situação não vai influenciar muito o resultado final do nosso objectivo uma vez que, a nossa zona de interesse não inclui os índices problemáticos.

A partir de agora estamos em condições de implementar o algoritmo iterativo através das

equações 4.10 e 4.11:

$$\mathbf{F}(e^{j\omega_q}) = MT_s\mathbf{H}^{-1}(e^{j\omega_q})\mathbf{T}(e^{j\omega_q}) \tag{4.10}$$

Onde ω_q contempla apenas as frequências correspondentes à zona de interesse definida por J, resultando assim um menor custo computacional (uma das vantagens deste algoritmo).

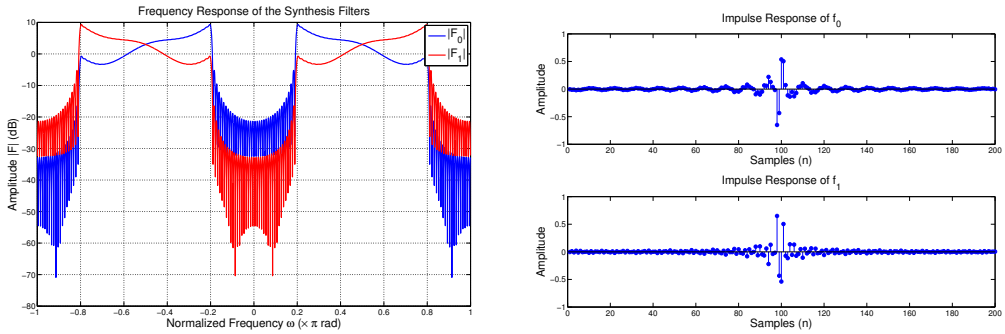
Para determinar a resposta impulsional dos filtros de síntese foi usada a DFT inversa.

$$\mathbf{F}_m = \mathbf{W}^H\mathbf{f}_m \tag{4.11}$$

O processo iterativo é implementado através do seguinte código matlab:

```
for ite=1:niter
    for w=J0p(1):J1p(end)
        % Forcing the Inversion in the Frequency Domain
        F(:,1,w)=H(:,:,w) \ T(:,1,w);
    end

    for w=J1n(1):J0n(end)
        % Forcing the Inversion in the Frequency Domain
        F(:,1,w)=H(:,:,w) \ T(:,1,w);
    end

    Ft=transpose(reshape(F,2,2*N));

    for k=1:M
        f(:,k)=real(ifft(Ft(:,k)));
        % Truncation in Time Domain
        f(L+1:end,k)=0;

        F(k,1,:)=fft(f(:,k),2*N);
    end
end
```

O resultado do processo iterativo pode ser observado através da figura da resposta em frequência (figura 4.14(a)) e da resposta impulsional (figura 4.14(b)) dos filtros inversos



(a) Resposta em frequência (b) Resposta impulsional

Figura 4.14: Resposta em frequência e resposta impulsional dos filtros de síntese

---

Como no caso de um filtro foi calculado o produto dos filtros de análises com os filtros de síntese para determinar o erro provocado pelo algoritmo

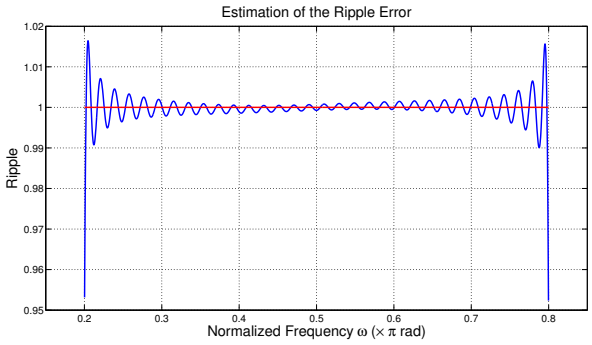
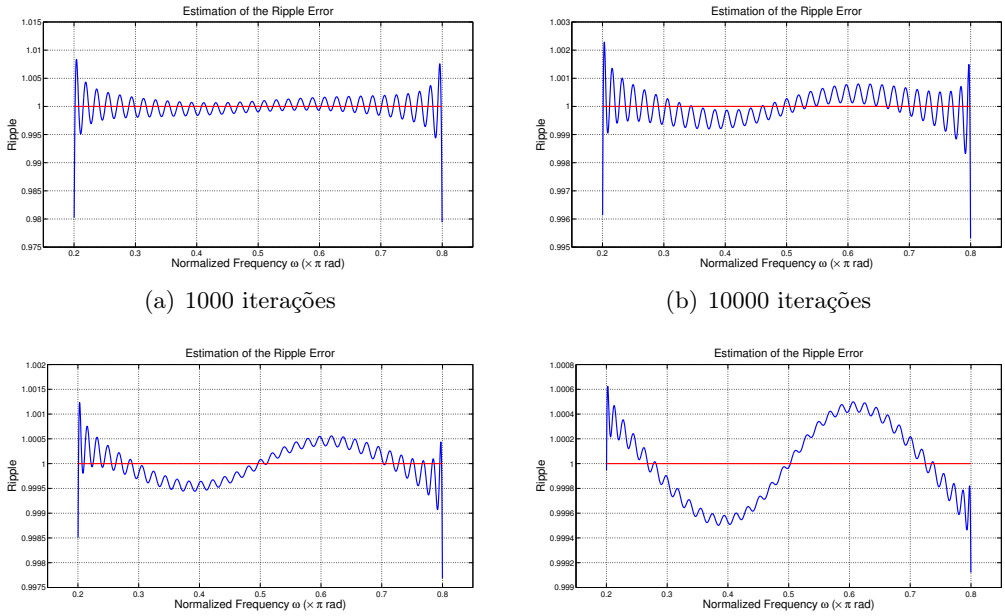


Figura 4.15: *Ripple* da inversão

Comparando várias simulações pode-se concluir que o *ripple* diminui com o aumento do número de iterações. Verifica-se também que o *ripple* é maior nas bandas de transição e que com o aumento para um número elevado de iterações começa a ocorrer um efeito de oscilação com amplitude na ordem dos 5×10^{-4}.



(a) 1000 iterações (b) 10000 iterações

(c) 100000 iterações (d) 1000000 iterações

Figura 4.16: Comparação das várias simulações com diferentes valores de iterações

---

É também possível concluir pela consulta da tabela 4.4 que o tempo de cálculo apresenta um aumento linear com o aumento do número de iterações, como acontecia no caso anterior de um filtro. Os valores foram obtidos com o uso do mesmo computador que na secção anterior.

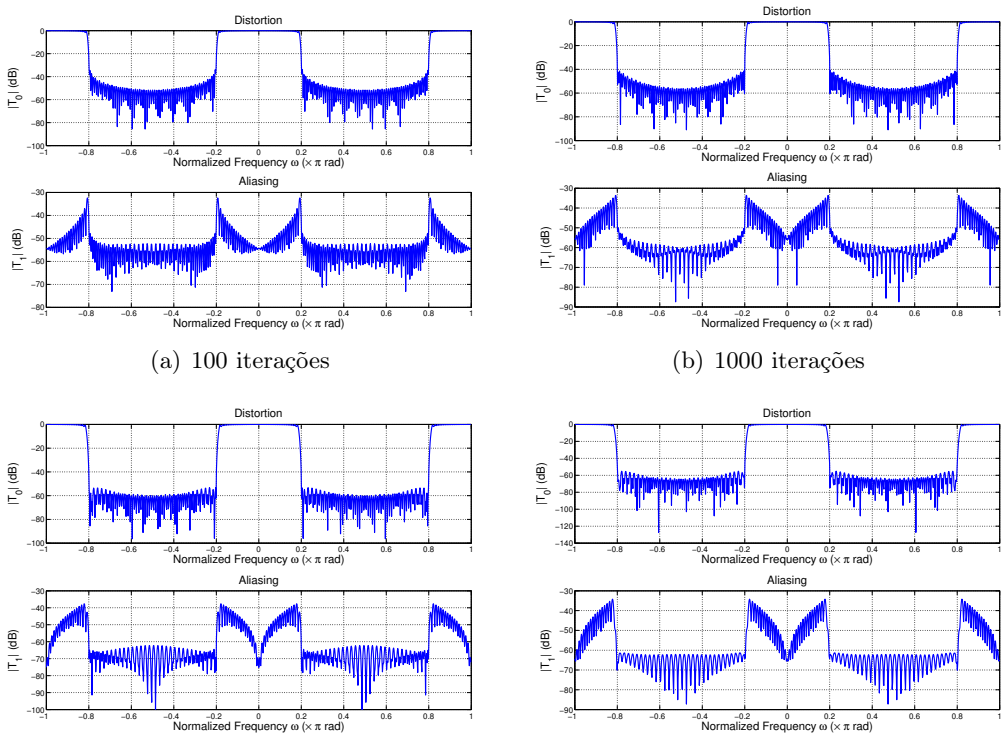
Niter	time
100	1.394048 segundos
1000	12.696590 segundos
1000	125.217890 segundos
10000	1242.352807 segundos

Tabela 4.4: Tempo de simulação do algoritmo Papoulis-Gerchberg aplicado a dois filtros com diferentes números de iterações

4.3.3 Simulações para diferentes números de iterações

Dos resultados obtidos neste capítulo, pode-se concluir então que o método baseado em POCS consegue inverter os filtros de análise dentro da banda de interesse com um *ripple* consideravelmente reduzido. Concluí-se, também, que este *ripple* pode ser diminuido através do aumento do número de iterações mas que por sua vez irá aumentar o custo computacional.

Calculando as componentes de distorção e *aliasing* para diferentes valores de iterações também é possível verificar que estas melhoram com o aumento do número de iterações



(a) 100 iterações

(b) 1000 iterações

(c) 10000 iterações

(d) 100000 iterações

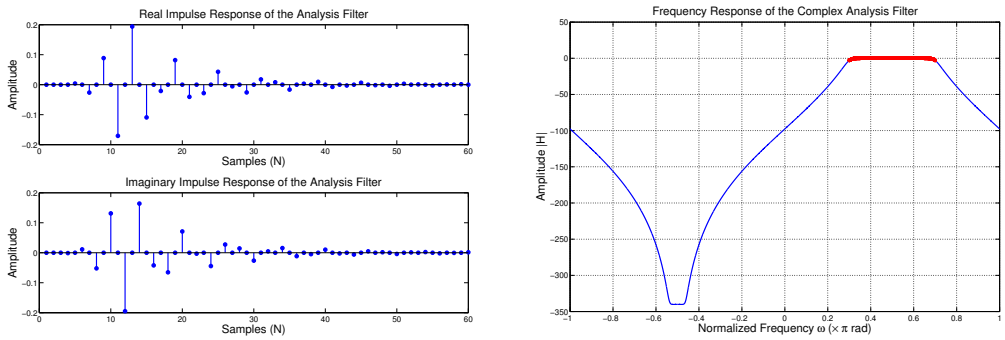
Figura 4.17: Distorção e Aliasing de várias simulações com diferentes valores de iterações

4.4 POCS aplicado a Filtros Complexos

É possível fazer uma abordagem comparativa entre os filtros reais e os filtros complexos, tendo em atenção as diferenças dos mesmos. A utilidade dos filtros complexos resume-se ao facto de permitirem trabalhar com sinais que não apresentem simetria par ao longo das frequências em banda-base. Outra diferença que é facilmente vísivel, é a resposta impulsional possuir componentes reais e complexas.

4.4.1 Um Filtro Complexo

Comparativamente ao filtros reais de síntese, o cálculo dos filtros complexos é menos exigente porque não é obrigatório existir simetria par, logo a zona de interesse ω_I fica reduzida a metade (figura 4.18(b)). Mas apresenta a necessidade de se efectuar o dobro dos cálculos no domínio do tempo porque os filtros possuem componentes reais e imaginárias de resposta impulsional (figura 4.18(a)).

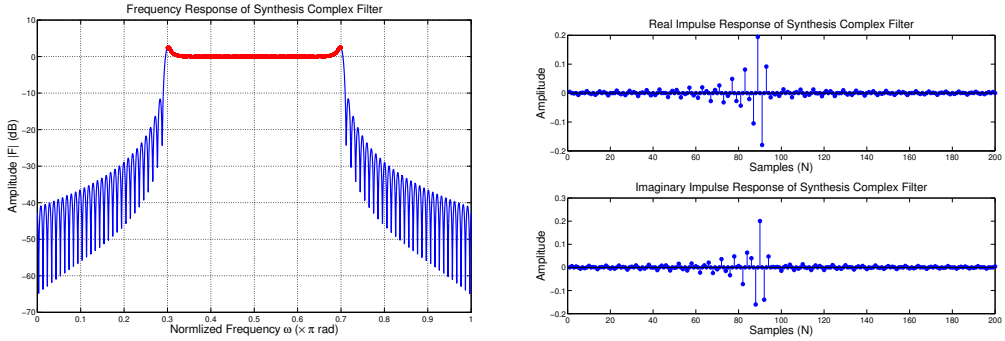


(a) Resposta impulsional (b) Resposta em frequência

Figura 4.18: Resposta impulsional e resposta em frequência do filtro de análise complexo

---

De modo a lidar com os filtros complexos é necessário proceder a alguns ajustes em relação ao algortimo de Papoulis-Gerchberg aplicado na secção 4.3. Durante o processo iterativo é forçada a inversão numa única zona de intersse ω_I e para efectuar a truncagem da resposta impulsional é necessário truncar a sua componente real e imaginária. Se este aspecto for desprezado, os filtros de síntese obtidos serão reais.



(a) Resposta em frequência (b) Resposta impulsional

Figura 4.19: Resposta em frequência e resposta impulsional do filtro de síntese

Mais uma vez foi também calculado o produto entre as rspostas em frequência de análise e de síntese, para averiguar o comportamento do *ripple*, resultante das iterações e que pode ser observado na figura 4.20:

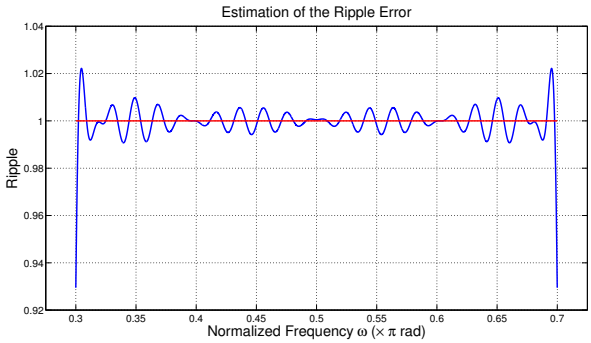


Figura 4.20: *Ripple* da inversão

Para uma melhor visualização do resultado final, na figura 4.21 são sobrepostas as respostas em frequência do filtro de análise e de síntese. Pode-se observar o comportamento do filtro de síntese que está bem definido na zona de interesse ω_I e o seu *ripple* na zona de não-interesse $\omega_N = \omega - \omega_I$.

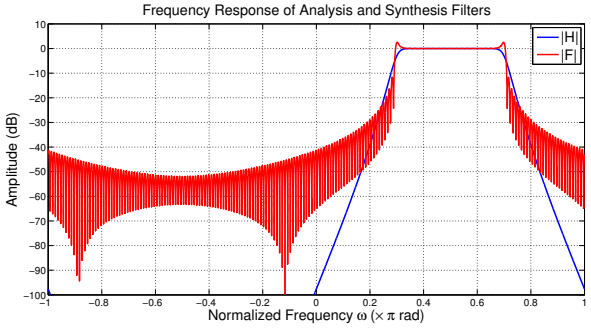
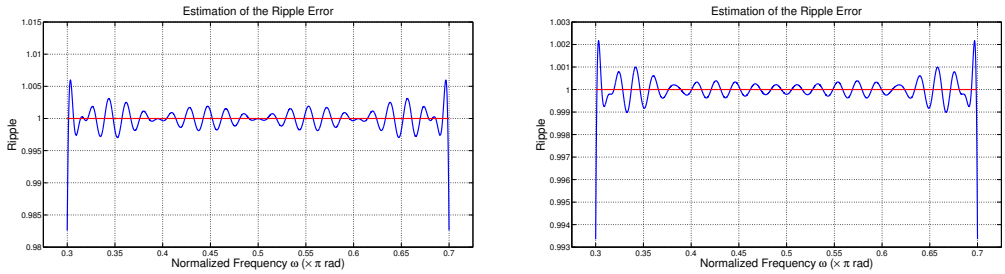


Figura 4.21: Comparação da resposta em frequência do filtro de análise e de síntese

4.4.2 Simulações para diferentes números de iterações

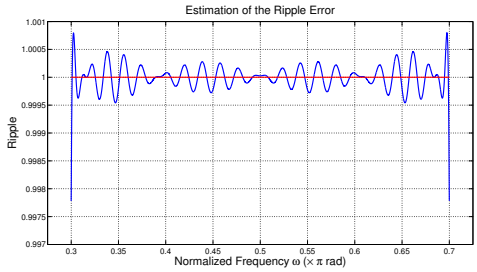
Se existir a necessidade de diminuir o *ripple* na zona de interesse ω_I é possível resolvê-la à custa do aumento do número de iterações. Note-se que este aumento causará uma maior complexidade nos cálculos efectuados, necessitando-se de mais tempo para efectuar os mesmos.

Será de bom senso encontrar um equilíbrio entre o *ripple* admitido e o número de iterações necessárias.

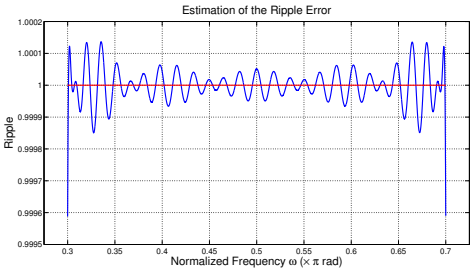


(a) 1000 iterações

(b) 10000 iterações



(c) 100000 iterações

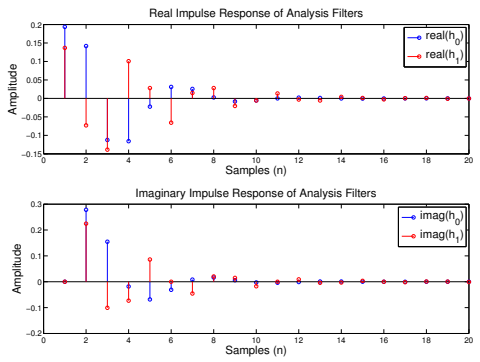


(d) 1000000 iterações

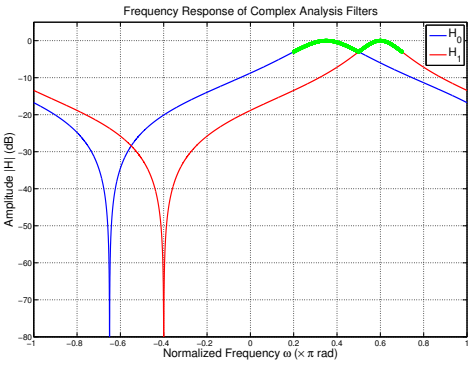
Figura 4.22: Erro de simulações para diferentes número de iterações

4.4.3 Dois filtros complexos

Nesta altura, resta-nos o estudo do caso de 2 filtros complexos. Mais uma vez foram considerados 2 filtros passa-banda de ordem 2. Na figura 4.23 estão representadas as respostas impulsional e em frequência.



(a) Resposta impulsional



(b) Resposta em frequência

Figura 4.23: Resposta impulsional e resposta em frequência dos filtros de análise complexos

Uma vez mais, é necessário verificar o comportamento do número de condição da matriz de análise de modo a demonstrar que do seu uso não irão resultar problemas em relação ao sistema de equações 3.7 utilizado para calcular as respostas em frequência dos filtros de síntese.

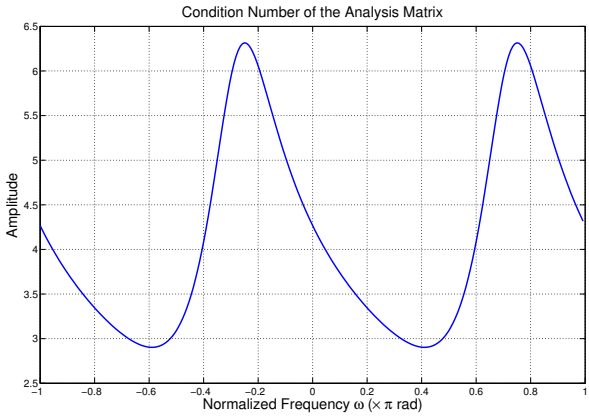
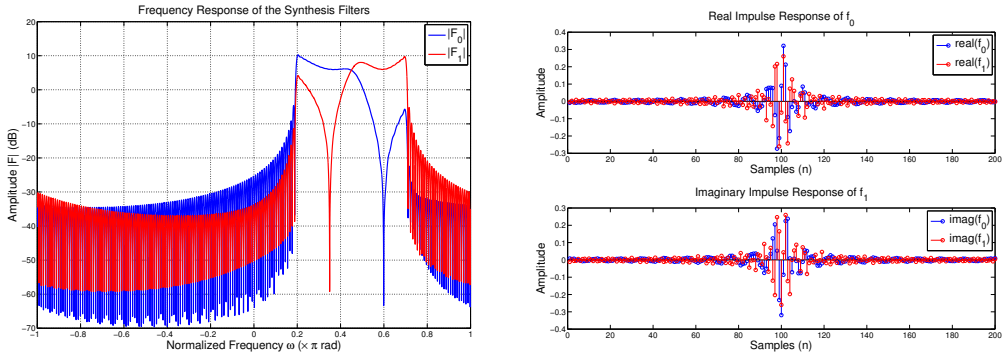


Figura 4.24: Número de condição da matriz de análise complexa

---

Pela observação da figura 4.24 conclui-se que o número de condição apresenta valores aceitáveis. Após feita esta consideração é então possível proceder ao cálculo da resposta em frequência dos filtros de síntese, que se pode observar na figura 4.25(a) e posteriormente ao cálculo da resposta impulsional na figura 4.25(b):



(a) Resposta em frequência

(b) Resposta impulsional

Figura 4.25: Resposta em frequência e resposta impulsional dos filtros de síntese complexos

De modo a confirmar o bom funcionamento do algoritmo, mais uma vez é colocado um impulso na entrada do banco de filtros de análise H_0 e H_1. Após a aplicação do algoritmo é calculada a distorção e o *aliasing* total gerado pelo sistema (figura 4.26)

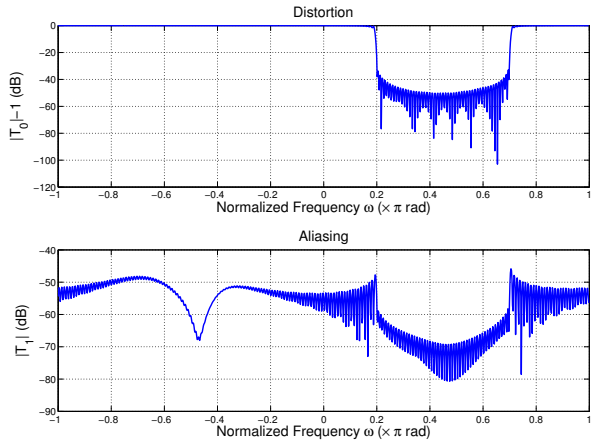


Figura 4.26: Distorção e *Aliasing*

Pode-se verificar na zona de interesse w_I, o sistema consegue atingir valores de distorção e *aliasing* na ordem dos -60 dB.

Capítulo 5

Sincronização

Neste capítulo irá ser testado o algoritmo desenvolvido para sincronizar os dois canais. Um primeiro teste já foi implementado com um sinal gerado em *Matlab* (discutido na secção 2.3). Após o correcto funcionamento deste teste passaremos ao uso do USRP para aquirir sinais RF e aplicar o mesmo algoritmo. Serão testados, numa primeira fase um sinal sinusóidal e em seguida um sinal com uma modulação Binary Phase Shift Keying (BPSK).

5.1 Universal Software Radio Peripheral

Nesta parte final do trabalho irá ser usado um Universal Software Radio Peripheral (USRP) N210 da *Ettus* (figura 5.1) que nos permitirá adquirir alguns sinais que possuem as não linearidades dos componentes reais.

Figura 5.1: USRP N210

https://www.ettus.com/product/details/UN210-KIT

Este USRP é constituído por uma *motherboard* que contém uma Full Programable Gates Array (FPGA) 3A-DSP 3400 da *Spartan*, possui duas ADCs de 14 bits a uma frequência de amostragem até 100 MS/s e duas DACs de 16 bits com uma frequência até 400 MS/s. Ainda na *motherboard* é possível programar os *downcoverters* e *upconverters* digitais. A programação da FPGA será realizada no GNURadio em sistema operativo *Linux* para tirar partido da maior estabilidade dos drivers USRP Hardware Driver (UHD). Para a comunicação entre o

computador e o USRP é usada a *interface* de *Gigabit Ethernet*.

O seu funcionamento será equivalente aos dois andares super-heteródinos. Serão os erros introduzidos pela implementação dos vários filtros e *phase-noise* da Phase Lock-Loop (PLL) que irão ser compensadas. Como é desejado estudar o funcionamento de dois andares super-heteródinos em cooperação é necessário usar o USRP juntamente com uma placa de expansão (*daughterboard*) funcionando como um Front-End.

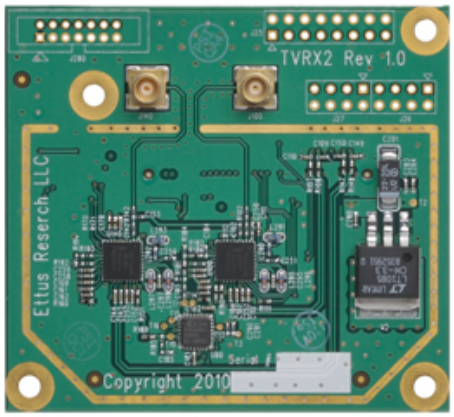


Figura 5.2: Placa de expansão TVRX2

https://www.ettus.com/product/details/TVRX2

Neste caso será a TVRX2 (figura 5.2) que permite receber dois canais independentes em simultâneo numa gama de frequências entre os 50 MHz e os 860 MHz com uma largura de banda na banda base de 10 MHz.

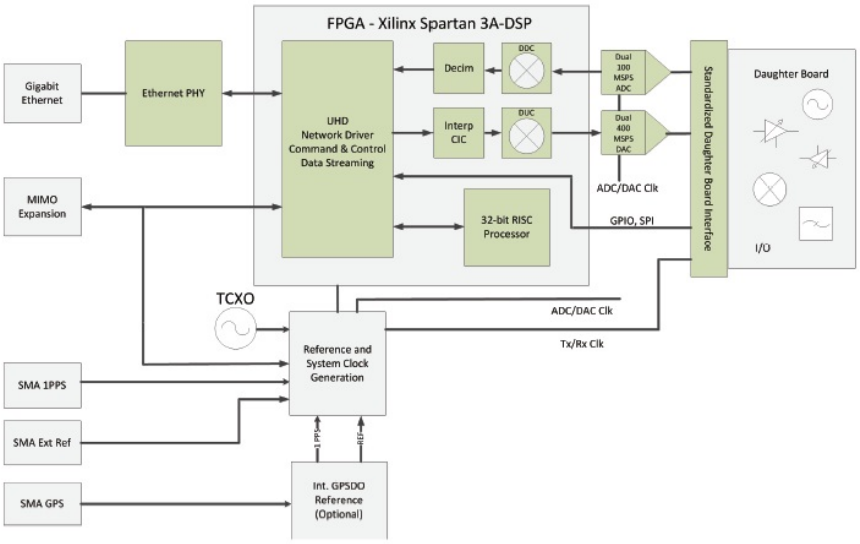


Figura 5.3: Diagrama de blocos do USRP N210

Na figura 5.3 encontra-se o diagram de blocos do USRP N210, em que as ADCs são implementadas através do uso do integrado *AZ62P44*, da *Texas Instrument* assim como as DACs são implementadas com o uso do integrado *AD9778*, da *Analog Devices* . O sinal de *clock* é gerado através no integrado *AD9510*, usando o oscilador interno (TCXO) ou uma fonte externa, também fabricado pela *Analog Devices*. Este mesmo *clock* é, também partilhado com a *daughterboard* e com os integrados das ADCs e DACs de modo a controlar a sua velocidade de operação. Após observação do espectro entre os 50 MHz e os 860 MHz, é possível verificar que são geradas componentes de ruído espaçadas de 16.(6) MHz, componentes estas, que são provocadas pelos circuitos geradores de *clock* implementados através do circuito *AD9510*.

Perante este factor foi considerada uma frequência de 550 MHz como a frequência central do nosso sistema. No caso da *daughterboard*, é constituída por 2 circuitos integrados *TDA18273*, produzidos pela ***NXP***, cujo diagrama de blocos pode ser observado na figura 5.4, que permitem efectuar a sintonização da zona do espectro que se pretende adquirir.

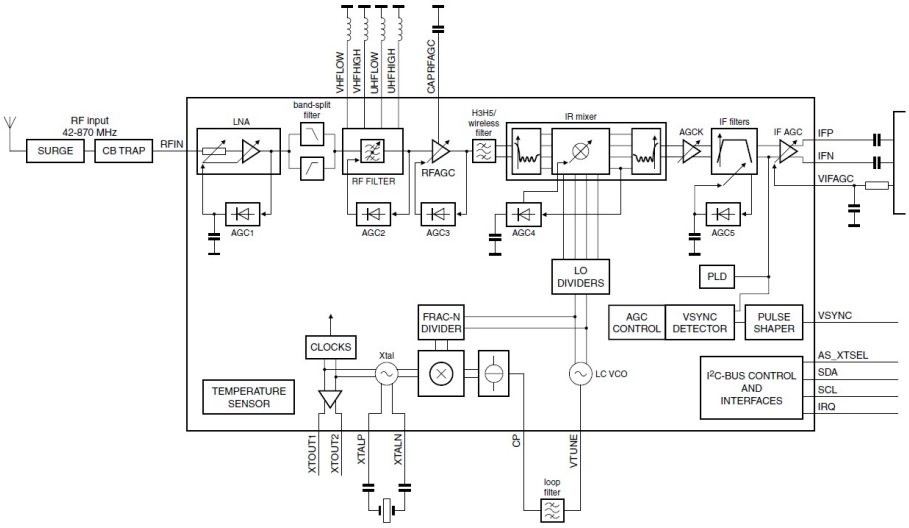


Figura 5.4: Diagrama de blocos do circuito integrado TDA18273

O sinal RF é encaminhado para um Low Noise Amplifier (LNA), sendo de seguida aplicado um filtro sintonizador RF para proteger o funcionamento do sintonizador contra componentes não desejadas. No próximo passo é efectuada a *down-conversion* para uma frequência intermédia. Na filtragem do sinal de IF é usado o conceito de Low Intermediate-Frequency (LIF), conceito este que implementa um filtro complexo e um filtro IF para se atingir uma melhor selectividade. O filtro IF é implementado através de um IF Filtro Passa-Baixo, um filtro IF *notch* e um IF Filtro Passa-Alto programável que permite obter uma maior flexibilidade. Todos os passos referidos podem ser controlados por *stepped automatic gain control* de modo a optimizar a Signal-to-Noise Ratio (SNR).

5.2 Aplicação a sinais de RF adquiridos através de USRP

Como já tinha sido referido no início deste capítulo iremos usar o USRP (figura 5.5) para fazer a aquisição de alguns sinais. Juntamente com o USRP iremos usar a placa de expansão ***TVRX2*** (figura 5.6) que permite receber dois canais independentes em simultâneo. Estes canais irão adquirir o sinal RF, convertendo-o para a banda-base e em seguida, efectuando a conversão para o domínio digital, simulando assim o receptor super-heteródino de dois canais representado na figura 2.1.

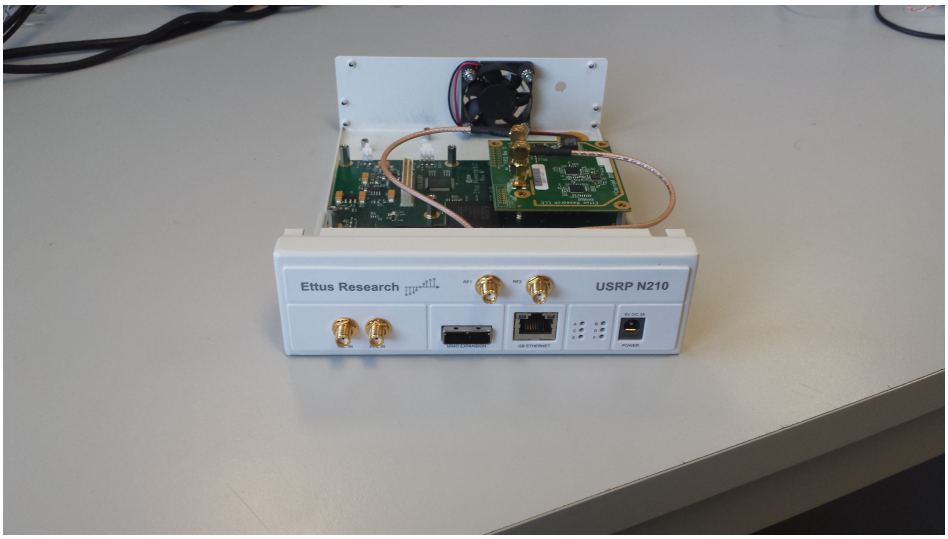


Figura 5.5: USRP

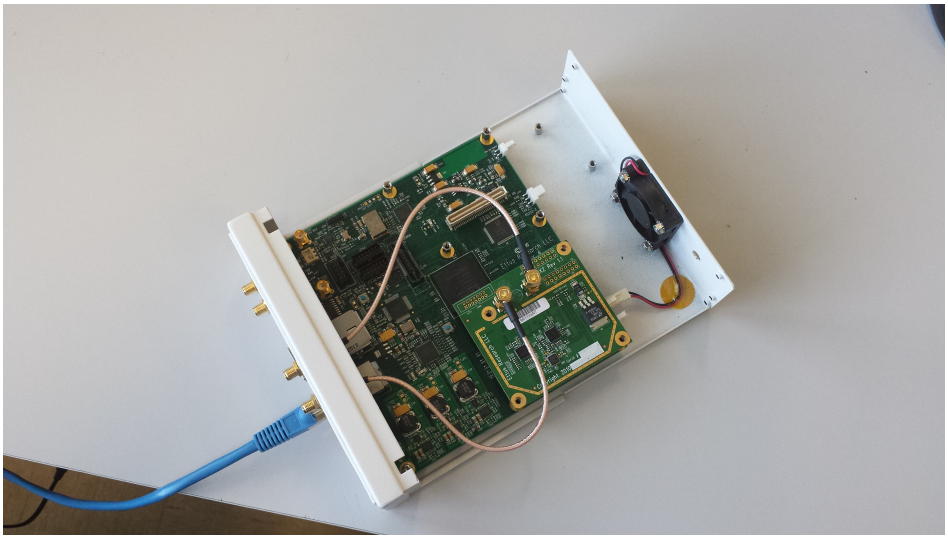


Figura 5.6: USRP com a placa de expansão TVRX2

Como o nosso objectivo é perceber como dois andares super-heteródinos adjacentes se comportam em trabalho cooperativo, iremos ter um único sinal Radio Frequency (RF) que é transmitido para os receptores do USRP através de um *power splitter* (figura 5.7) que separa o sinal em dois sinais identicos. Devido a limitações de material no laboratório foi usado um *splitter* de 1:3, em que foram usadas as portas 1 e 3 para enviar os sinais às entradas do USRP sendo necessário conectar uma carga de 50 Ω para evitar consequentes reflexões. O *splitter* usado foi o ZB3PD-63-S+ da *Mini-Circuits* que pode ser usado numa gama de frequências entre os 150 MHz e os 6 GHz, com uma *isolation* entre portas de 20 dB e uma *phase unbalance* máxima de 5º.

Figura 5.7: Splitter ZB3PD-63-S+ da *Mini-Circuits*

5.2.1 Programação do USRP

Para a utilização do USRP como dois receptores heteródinos foi criado um programa em línguagem *Python* no *Software* GNURadio, que pode ser observado na figura 5.8 onde se usa um bloco de USRP como *source*. Este bloco é programado para dois canais que gravam a sua saída nos ficheiros ch1.bin e ch2.bin. Cada canal é programado para a sua frequência central controlada pelas variáveis $F1$ e $F2$ que irão ser iguais a 549.5 MHz e 550.5 MHz, respectivamente. A variábel BW controla a largura de banda dos filtros implementados pelo USRP sendo imposta com o valor de 1 MHz. Para se conseguir adquirir os dois sinais é necessário impor uma frequência de amostragem que seja no mínimo o dobro da largura de banda, sendo por isso definida a 2 MS/s. Foi ainda introduzida a variável ADC_Gain para permitir controlar o ganho das ADCs durante a simulação. Os outros blocos visíveis são ferramentas gráficas que nos permitem observar em tempo real o que está a ser adquirido, tanto no domínio do tempo como na frequência.

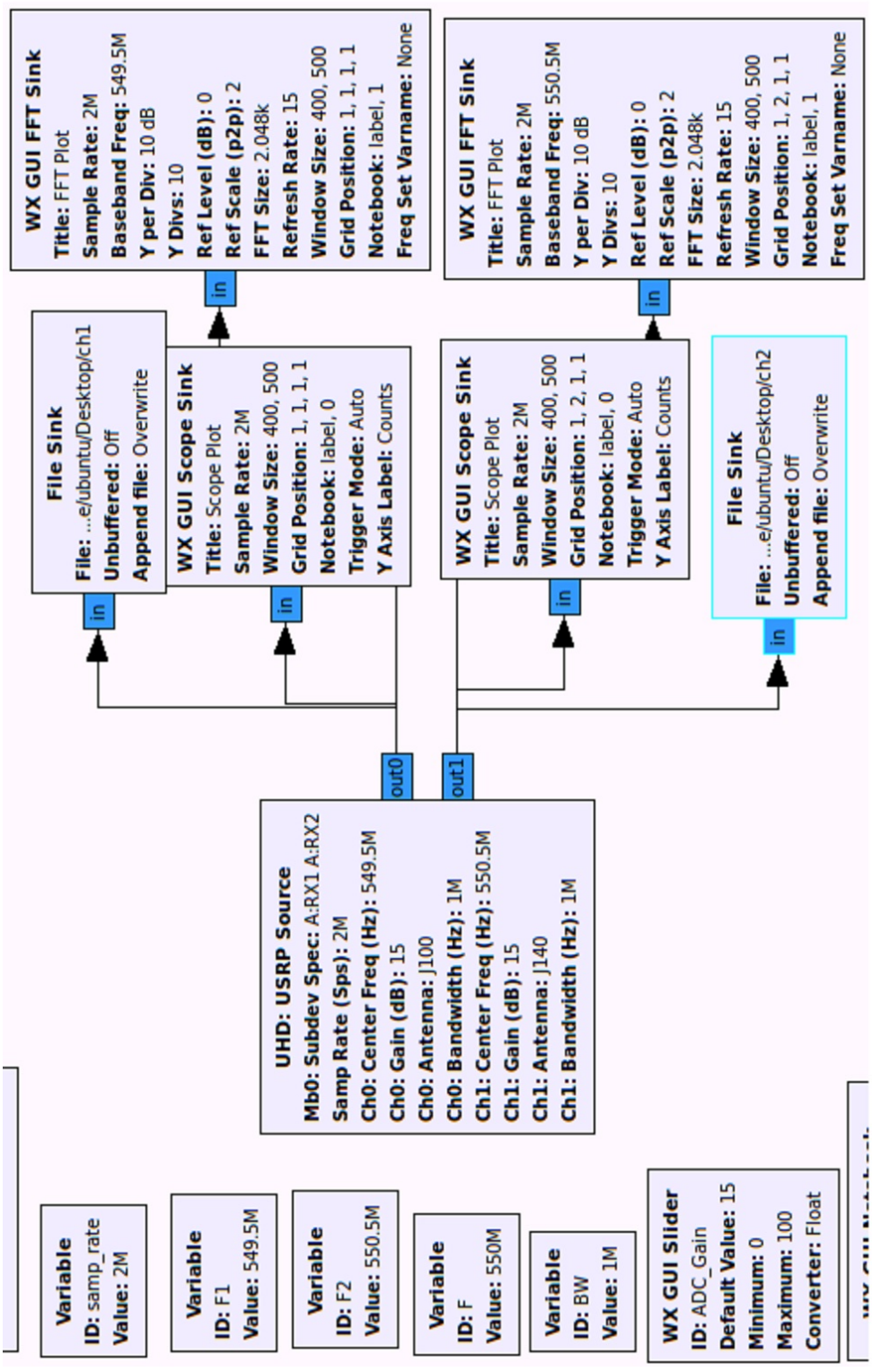


Figura 5.8: Programa em GNURadio para aquisição de um sinal usando o USRP

5.2.2 Aquisição de sinais

5.2.2.1 Aquisição de uma Sinusóise

Numa primeira fase foi gerada uma sinusóide com uma frequência de 550 MHz através do uso do gerador da *Rohde & Schwarz SMR40* (figura 5.9) com uma potência de saída de -3 dBm.

Figura 5.9: Gerador R&S SMR40

Este sinal é então enviado para um *power splitter* que fará a ligação com os dois canais do USRP como pode ser observado na figura 5.10

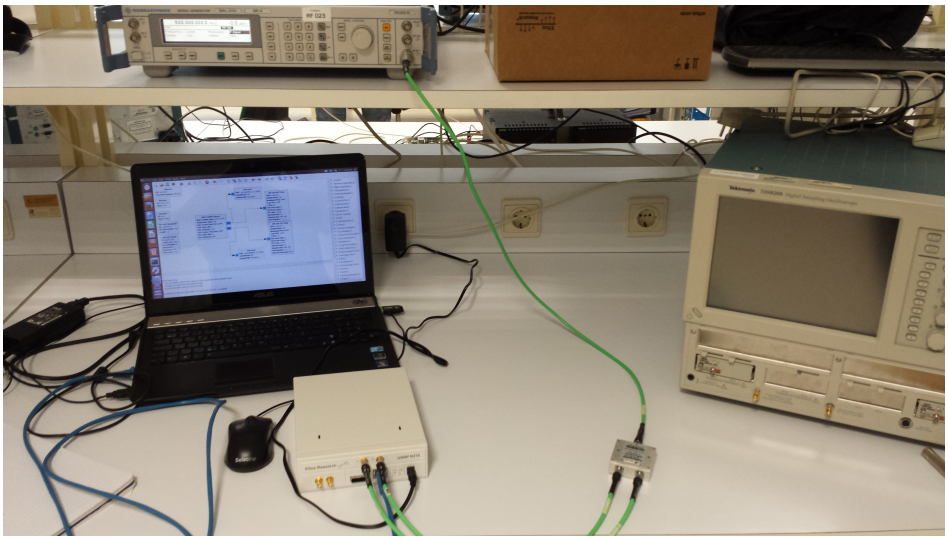


Figura 5.10: Montagem

Os sinais em banda-base aquirido pelo USRP são guardados em ficheiros .bin que serão usados à postiriori para o processamento em *Matlab*.

5.2.2.2 Aquisição de um sinal BPSK

Para gerar um sinal BPSK foi usado o gerador *SMW200A* (figura 5.11) também da *Rohde & Schwarz*. Este sinal foi gerado com uma frequência RF de 550 MHz com uma potência de saída de -10 dBm. Foi imposta uma sequência de bits pré-definida como '0000 0001 0010 0011 0101 0111 0111 0000 0010 0100 0110 1111 1001' com uma frequência de 500 Ksym/seg.

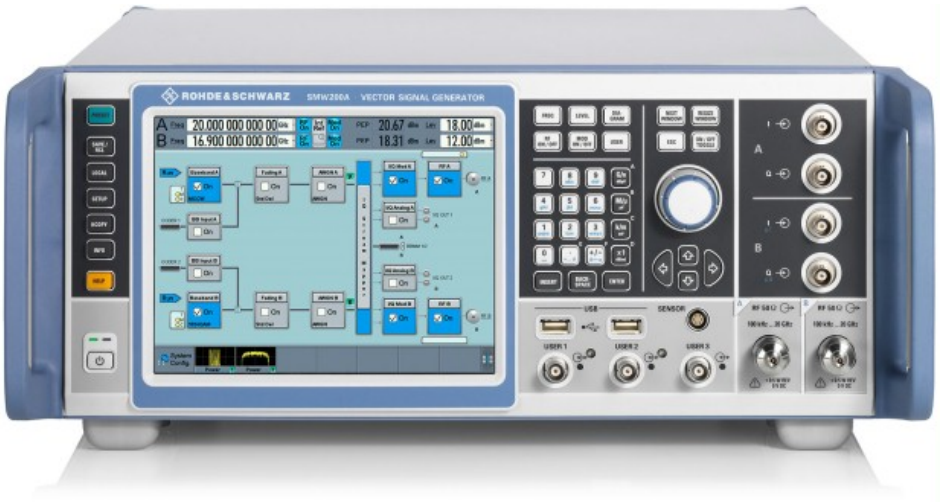


Figura 5.11: Gerador R&S SMW200A

http : *//www.rohde* − *schwarz.com/en/product/smw*200*a* − *productstartpage*$_6$3493 − 38656.*html*

Antes de criar os ficheiros .bin foram observados os espectros do sinal recebido pelo USRP (figura 5.12) e gerado pelo gerador (figura 5.13)

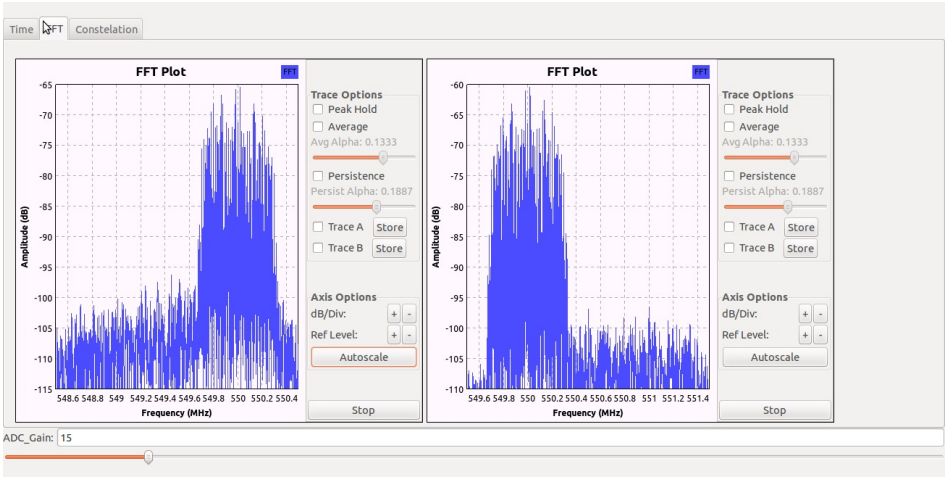


Figura 5.12: Aquisição do sinal em GNURadio

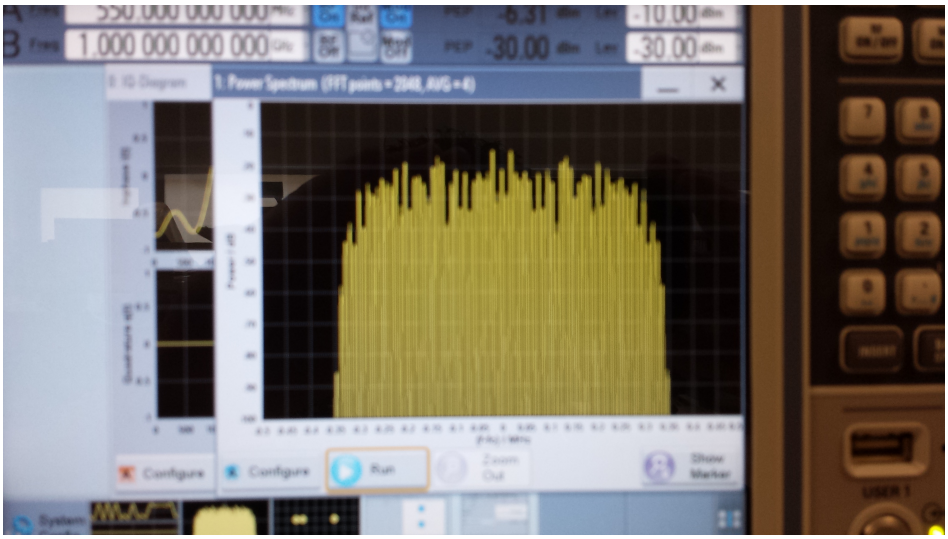


Figura 5.13: Visualização do espectro de frequências do sinal gerado

5.2.3 Aplicação do algoritmo

Foi desenvolvido um pequeno código que interpreta os ficheiros binários gerados no *Software* GNURadio como contendo amostras com uma determinada frequência de amostragem e tendo em conta que os sinais adquiridos possuíam uma modelação IQ, significa que as amostras de índice ímpar correspondem às amostras em quadratura enquanto que as amostras de índice par correspondem às amostras em fase. A função desenvolvida para carregar os ficheiros .bin foi designada por **readGNURadio** e recebe como parâmetro de entrada o nome do ficheiro .bin retornando um vector complexo. Após a realização deste passo é possível aplicar directamente o algoritmo de correlação para calcular o desfasamento entre o canal 1 e o canal 2 adquiridos pelo USRP.

5.2.3.1 Aplicação de algoritmo ao sinal sinúsoidal adquirido

Nas figuras 5.14 e 5.15 podem-se obeservar os sinais x_1 e x_2 provinientes do sinal sinusóidal aquirido, no domínio do tempo e frequência, respectivamente:

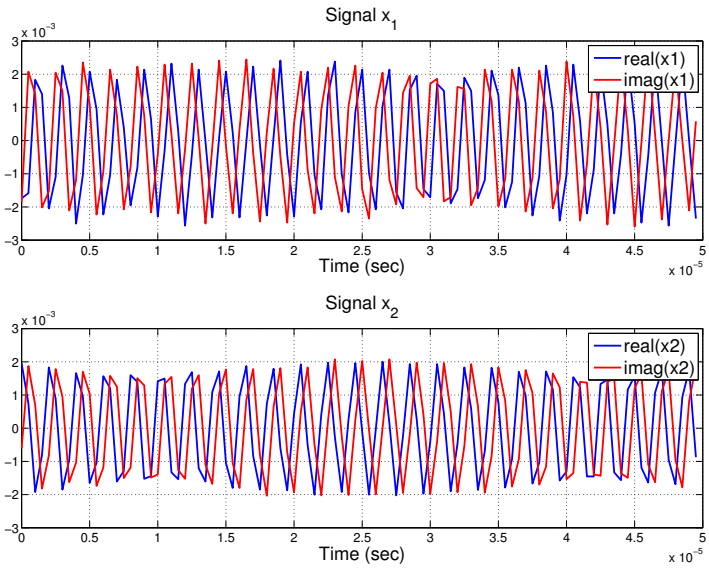


Figura 5.14: Sinais x_1 e x_2 adquiridos pelo USRP

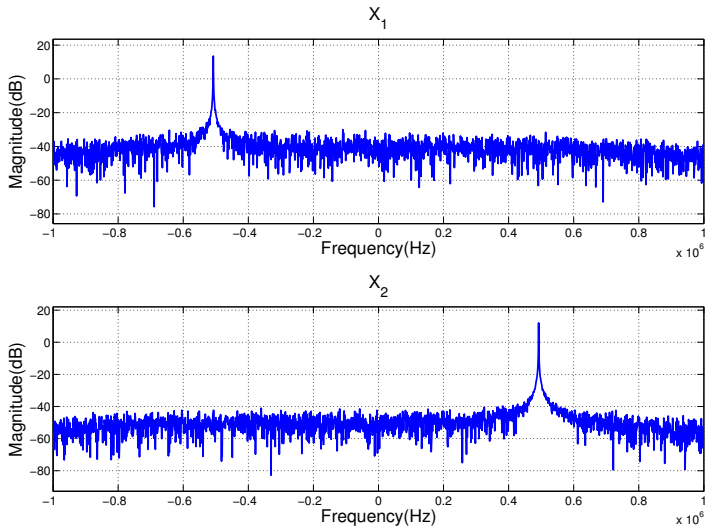


Figura 5.15: Resposta em frequência dos sinais x_1 e x_2

Com a aplicação do bloco de síntese descrito na figura 2.7 obtém-se os sinais y_1 e y_2 representados nas figuras 5.16 e 5.17:

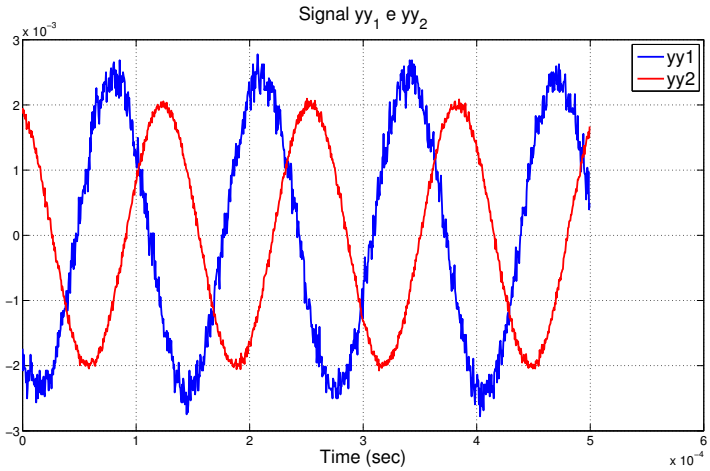


Figura 5.16: Sinais y_1 e y_2, no domínio do tempo, à saída do banco de síntese

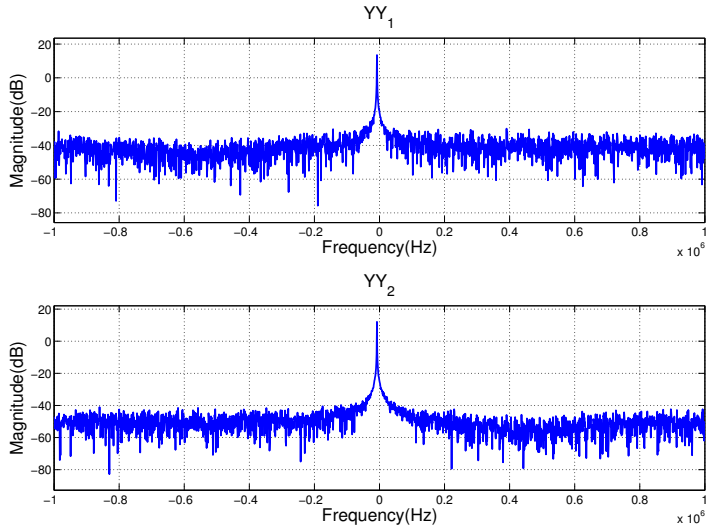


Figura 5.17: Resposta em frequência dos sinais y_1 e y_2 à saída do banco de síntese

O sinal y obtem-se através da soma de y_1 e y_2:

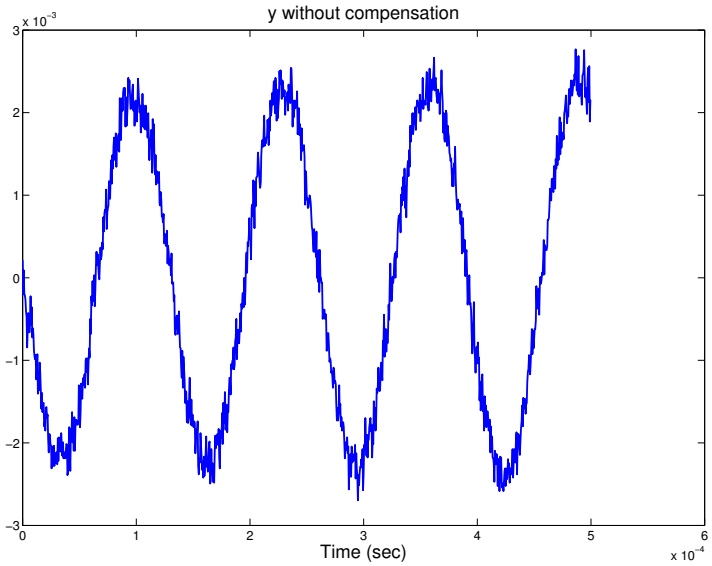


Figura 5.18: Sinusóide reconstruída sem compensação

Se os sinais estivesse-se sincronizados, a sua soma iria originar uma sinusóide com o dobro da amplitude dos sinais y_1 e y_2, o que visivelmente não acontece (figura 5.18).

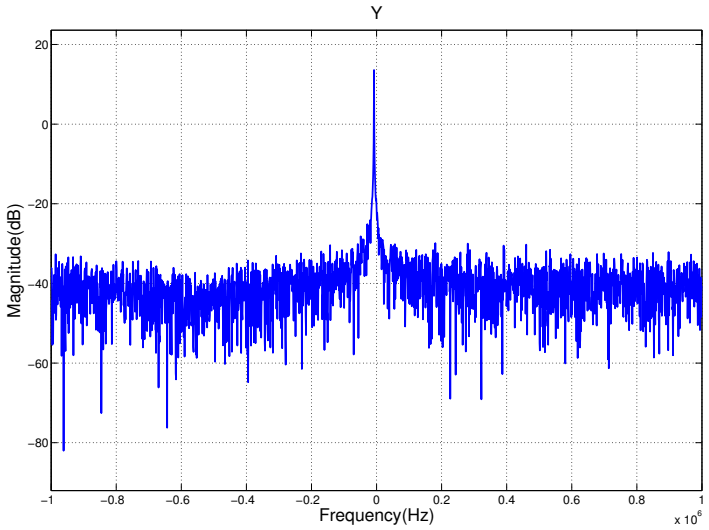


Figura 5.19: Resposta em frequência da sinusóide reconstruída sem compensação

Será então aplicado o algoritmo descrito na secção 2.2 em que será efectuada, em primeiro lugar, a correlação na frequência, de modo a corrigir o desvio de frequência entre os dois canais e, em segundo, a correlação no tempo para permitir a correcção de fase entre os mesmo.

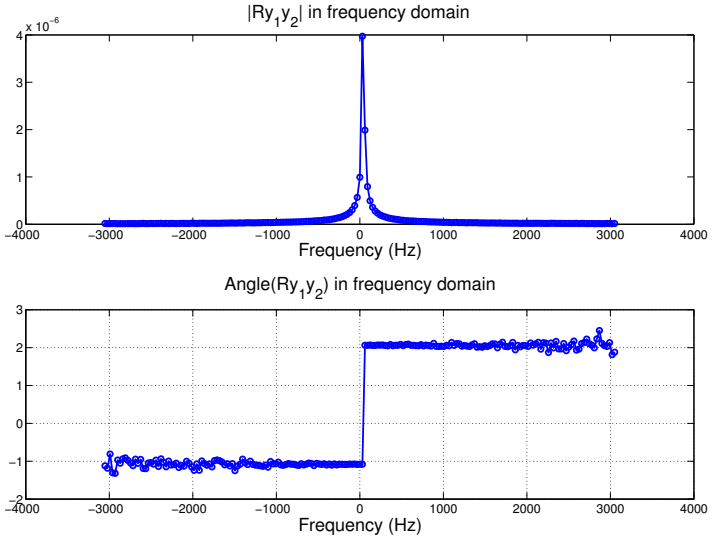


Figura 5.20: Correlação na frequência

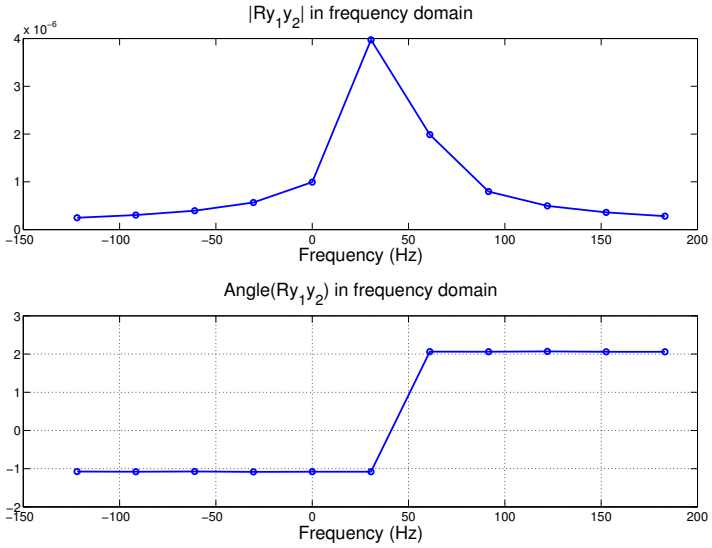


Figura 5.21: Zoom da correlação na frequência

Os resultados da correlação na frequência são apresentados na figura 5.20, com a zona do máximo ampliada na figura 5.21. É calculado que o desvio de frequência é apróximadamente de 30.5 Hz.

```
c_Delta =

   30.5176

Frequency compensation of: 30.5176 Hz
```

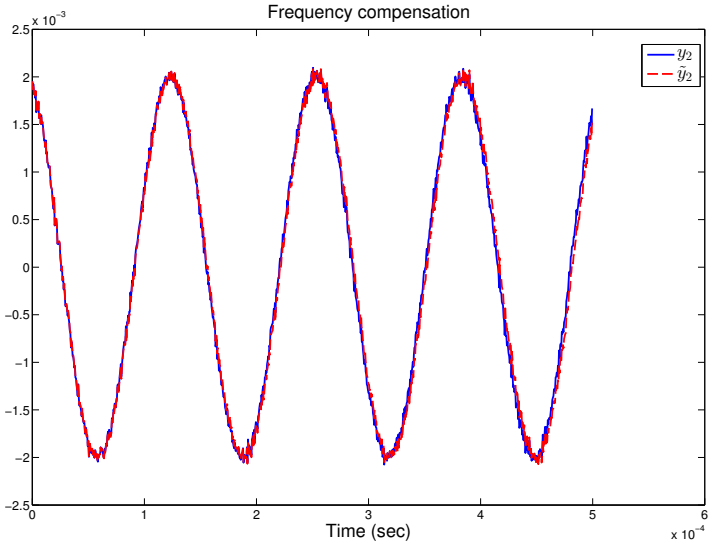


Figura 5.22: Compensação na frequência

Após a devida compensação da frequência é então procedida à correlação no domínio do tempo para determinar qual a diferença de fase entre os dois canais, para poderem ser correctamente adicionadas. A forma da correlação pode ser observada na figura 5.23. Note-se que o resultado da correlação entre dois sinais sinusóidais é um sinal sinusóidal, onde o valor máximo da parte real equivale ao ponto onde os dois sinais se encontram alinhados.

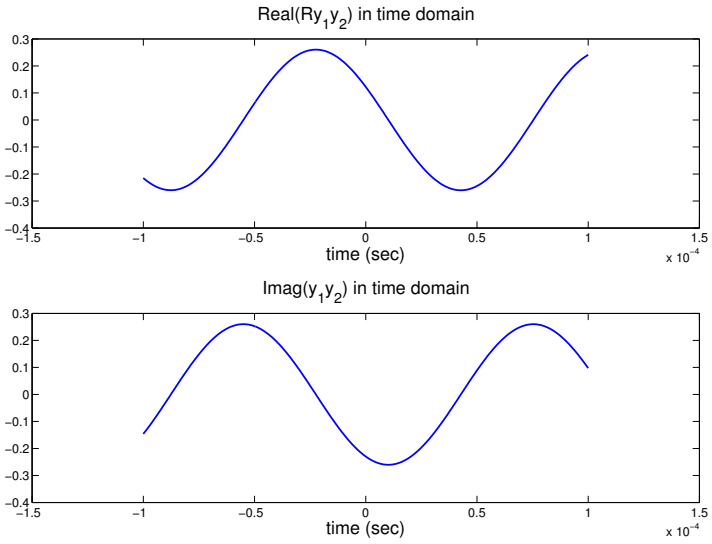


Figura 5.23: Correlação no tempo

Através do cálculo do índice do máximo da parte real é determinado qual a diferença de fase entre os dois sinais:

```
Desfazamento (rad): -1.1045
Desfazamento (º): -63.2813
```

Depois de realizada a devida compensação de fase é, então, possível reconstruir correctamente o sinal de entrada original (figuras 5.24 e 5.25).

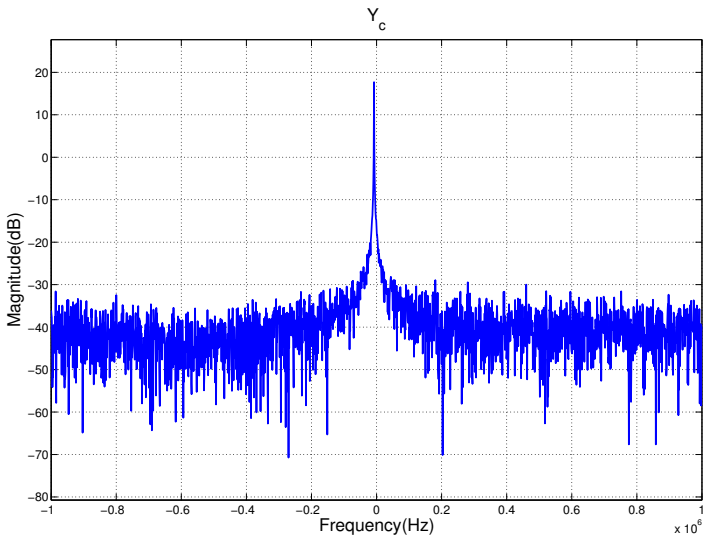


Figura 5.24: Resposta em frequência da sinusóide reconstruída

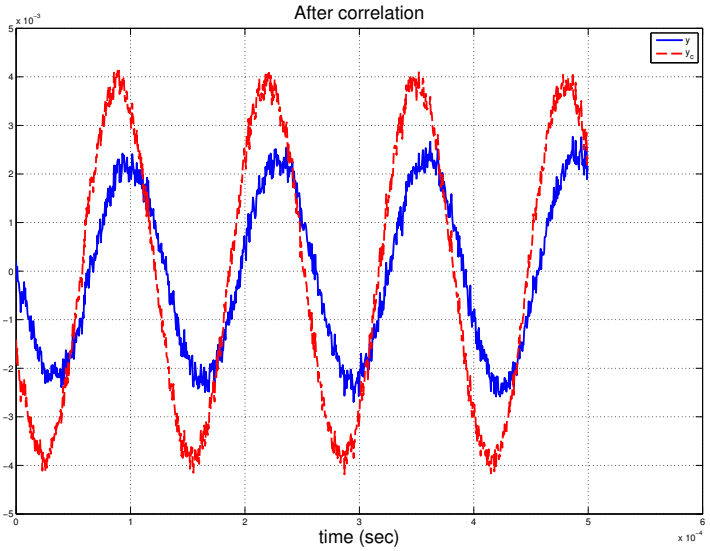


Figura 5.25: Sinusóide reconstruída

Na figura 5.25 é possível observar que o sinal reconstruído apresenta uma aplitude na ordem dos 4 mW, o dobro dos sinais y_1 e y_2, devido à compensação efectuada.

5.2.3.2 Aplicação de algoritmo ao sinal BPSK adquirido

Atraves da figura 5.26 é possível observar os sinais adquiridos pelo USRP:

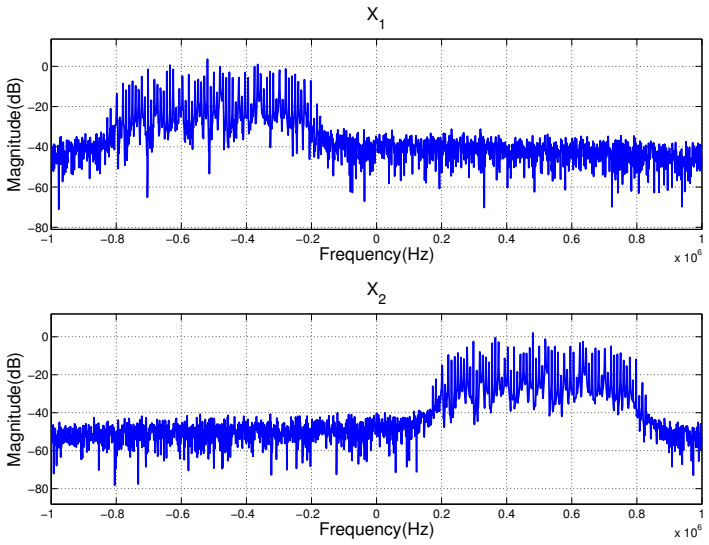


Figura 5.26: Resposta em frequência dos sinais x_1 e x_2

Nesta altura, estamos em condições de aplicar o algoritmo de correlação para reconstruir o sinal originalmente recebido. Na figura 5.27 é representada a operação de deslocação na frequência dos sinais para as respectivas frequências centrais para poderem ser somados, ou seja, é aplicado o banco de síntese:

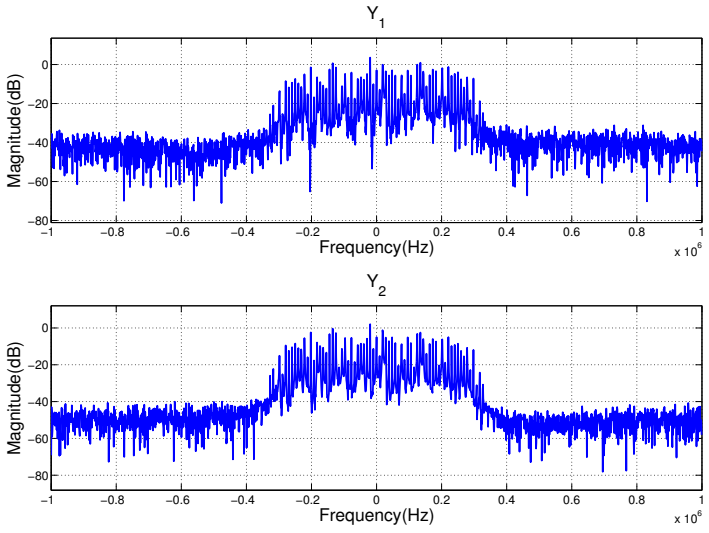


Figura 5.27: Resposta em frequência dos sinais y_1 e y_2

Através da correlação, numa primeira fase na frequência, e após devida compensação uma nova correlação, desta feita, no domínio do tempo, são obtidas as figuras 5.28 e 5.29, respectivamente:

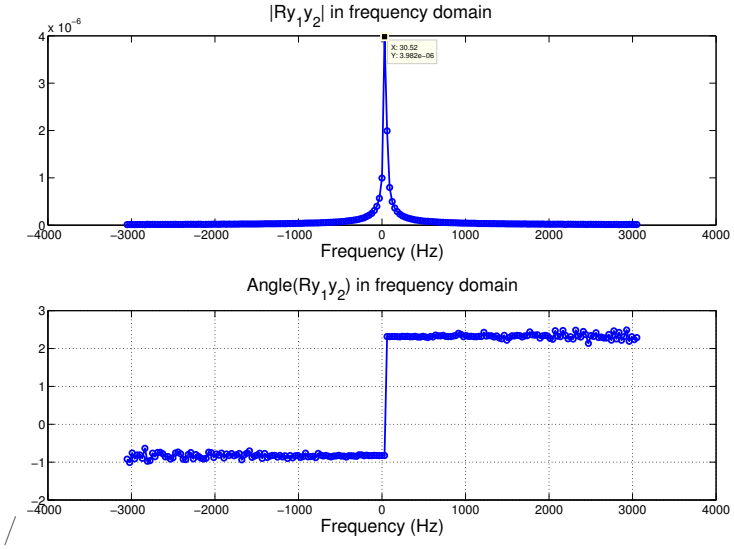


Figura 5.28: Correlação na frequência

```
Frequency compensation of: 30.5176 Hz
```

Após a compensação na frequência é realizada a correlação no tempo, para se determinar qual a diferença de fase:

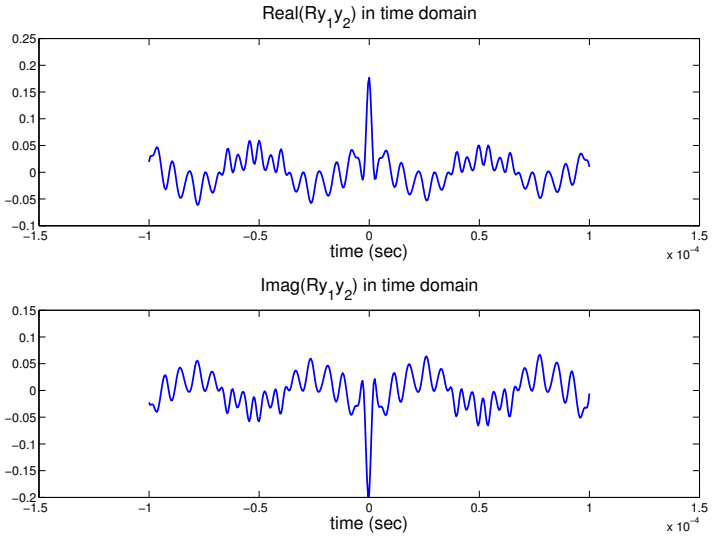


Figura 5.29: Correlação no tempo

No caso concreto de um sinal BPSK, é verificado que não existe diferença de fase entre os dois canais, uma vez que o máximo é calculado quando os sinais se encontram sobrepostos.

```
Desfazamento (rad): 0
Desfazamento (º): 0
```

O sinal reconstruído com compensação pode ser observado no domínio da frequência na figura 5.30

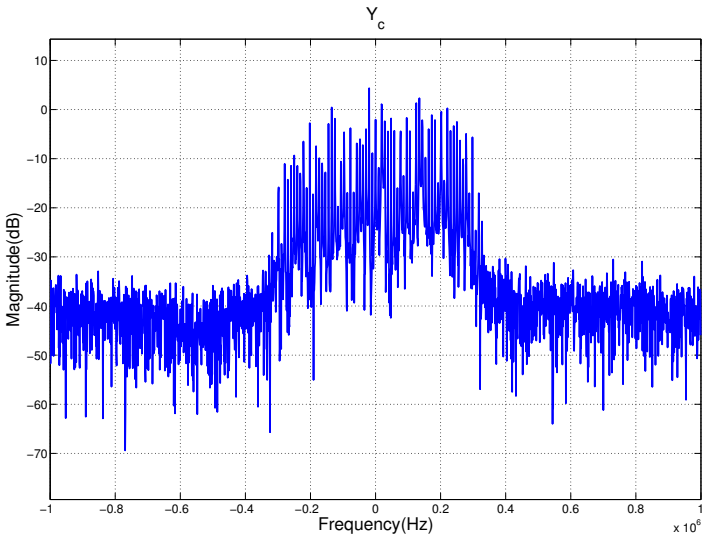


Figura 5.30: Sinal reconstruido com compensação

Pode-se concluir, então, que é possível sincronizar os dois canais de recepção heteródinos implementados no USRP, permitindo que seja possível

Capítulo 6

Conclusão

Após a realização deste trabalho foi conseguiu-se atingir os objectivos propostos de obter dois andares heteródinos a funcionar de um modo síncrono. Foi desenvolvido um algoritmo que recebe dois ficheiros binários, respectivos aos dois canais heteródinos programados no USRP e consegue determinar qual o desvio de frequência causado pela operação de heteródinagem prodecendo de seguida à sua compensação para finalmente poderem ser somados para reconstruir o sinal que se deseja adquirir.

Um dos próximos passos de estudo a realizar nesta área poderá ser o estudo da influência que um possível erro de fase poderá causar na realização da reconstrução do sinal em domínio digital. Um outro ponto de interesse, seria estudar o método de compensação de fase usando uma Phase Lock-Loop (PLL), ao invés da correlação como foi realizada neste trabalho. Para se concretizar um sistema completo é necessário proceder à caracterização dos filtros implementados no USRP de modo a que seja possível equalizá-los, usando os algoritmos desenvolvidos nos capítulos 3 e 4, de modo a que a resposta total do sistema seja plana. Num estado mais avançado, seria possível desenvolver um sistema de sincronização em tempo real, baseado numa FPGA e DSP que aplica-se o algoritmo desenvolvido neste trabalho.

Bibliografia

[1] D. F. Albuquerque, S. T. Soldado, J. M. N. Vieira, and N. B. Carvalho. Cochlear Radio. (September):209–212, 2010.

[2] D. F. Albuquerque, J. M. N. Vieira, N. B. Carvalho, and J. R. Pereira. Analog Filter Bank for Cochlear Radio. *2010 IEEE International Microwave Workshop Series on RF Front-ends for Software Defined and Cognitive Radio Solutions (IMWS)*, pages 1–4, February 2010.

[3] D. Asemani and J. Oksman. Influences of oversamplig and analog imperfections on Hybrid Filter Bank A/D converters.

[4] D Asemani, J Oksman, and P Duhamel. Subband architecture for Hybrid Filter Bank A / D converters. X(X):1–10, 2008.

[5] P. Cruz, N. B. Carvalho, and K. A. Remley. Designing and Testing Software-Defined Radios. pages 83–94, 2010.

[6] P. J. Ferreira. Interpolation and the Discrete Papoulis-Gerchberg Algorithm. *IEEE Transactions on Signal Processing*, 42, 1994.

[7] C. J. Galbraith, Student Member, R. D. White, L. Cheng, K. Grosh, and G. M. Rebeiz. Cochlea-Based RF Channelizing Filters. 55(4):969–979, 2008.

[8] R. Gomez-Garcia, J. Vieira, N. B. Carvalho, and J. P. Magalhaes. Mixed-domain receiver architecture for white space software-defined radio scenarios. *2012 IEEE International Symposium on Circuits and Systems*, pages 822–825, May 2012.

[9] Deepak Gopi. *Digital Front End for Base-station RF*. PhD thesis, 2011.

[10] S. Haykin. *CommunicationSystems*. Fourth edition.

[11] Rajesh Inti. *INVESTIGATION OF HYBRID FILTER BANK BASED ANALOG-TO-DIGITAL CONVERSION*. PhD thesis, India Institute of Technology, 2007.

[12] J. Kirkhorn. Introduction to IQ-demodulation of RF-data. 1999.

[13] P. Kiss, V. Prodanov, and J. Glas. Complex Low-Pass Filters. 2.

[14] P. Kiss, V. Prodanov, and J. Glas. COMPLEX LOW -PASS FILTERS. (May):2–5, 2002.

[15] C. Lelandais-Perrault, T. Petrescu, D. Poulton, P. Duhamel, and J. Oksman. Wideband, Bandpass, and Versatile Hybrid Filter Bank A/D Conversion for Software Radio. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(8):1772–1782, August 2009.

[16] P. Löwenborg, H. Johansson, and L. Wanhammar. A survey of filter bank A/D converters.

[17] P. Lowenborg, H. Johansson, and L. Wanhammar. Two-channel hybrid analog/digital filter banks with alias-free subbands. *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems (Cat.No.CH37144)*, 3:1162–1165.

[18] R. Lyons. Quadrature Signals : Complex, But Not Complicated. (November), 2008.

[19] R. G. Lyons. *Understanding Digital Signal Processing.pdf*.

[20] J. P. Magalhaes, T. Monteiro, J. M. N. Vieira, R. Gomez-Garcia, and N. B. Carvalho. Papoulis-Gerchberg Hybrid Filter Bank receiver for cognitive-/Software-Defined Radio systems. *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 69–72, May 2013.

[21] J. P. Magalhaes, J. M. N. Vieira, and R. Gomez-garcia. RF and IF Channelizers for Wide-Band Sensing In Cognitive / Software-Defined-Radio Receivers. pages 1158–1161, 2012.

[22] D. Manolakis and V. Ingle. *APPLIED DIGITAL SIGNAL PROCESSING*. Cambridge University Press.

[23] K. Martin. Complex Signal Processing is Not-Complex. 1981.

[24] U. Mengali and A. D'Andrea. *Synchronization Techniques for Digital Receivers (Applications of Communications Theory)*. 1997.

[25] H. Meyr, M. Moeneclaey, and S. A. Fechtel. *DIGITAL COMMUNICATION RECEIVERS - Synchronization, Channel Estimation and Signal Processing*.

[26] J. Mitola. The Software Radio Architecture. *IEEE Communications Magazine*, pages 26–38, 1995.

[27] J. Mitola. *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, 2000.

[28] J. Mitola and G. Maguire Jr. Cognitive Radio : Making Software Radios More Personal. (August):13–18, 1999.

[29] S. K. Mitra. *Digital Signal Processing - Computer Based Approach*.

[30] D. Navakauskas and A. Serackis. *DIGITAL SIGNAL PROCESSING TOOLS*. 2012.

[31] Z. Nikolova, G. Iliev, M. Ovtcharov, and V. Poulkov. Complex Digital Signal Processing in Telecommunications. 1999.

[32] Z. Nikolova, G. Stoyanov, G. Iliev, and V. Poulkov. Complex Coefficient IIR Digital Filters. pages 209–240, 2003.

[33] P. Oliveira and L. Gomes. Interpolation of signals with missing data using Principal Component Analysis. *Multidimensional Systems and Signal Processing*, 21(1):25–43, April 2009.

[34] A. V. Oppenheim. *Digital Signal Processing.*

[35] T. Petrescu, J. Oksman, P. Duhamel, and C. Lss. Synthesis of Hybrid Filter Banks by Global Frequency Domain Least Square Solving. (3):5565–5568, 2005.

[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes - The Art of Scientific Computing.* Cambridge University Press, third edition.

[37] J. G. Proakis. *Contemporary Communication Systems using Matlab - Proakis and Salehi.*

[38] J. G. Proakis. *Digital Communications.*

[39] J. G. Proakis. *Digital Signal Processing.* Fourth edition.

[40] J. G. Proakis. *DIGITAL SIGNAL PROCESSING using Matlab.* Third edition.

[41] J. G. Proakis. *Communications Systems Engineering.* 2001.

[42] T. O. Silva. *Apontamentos de Processamento Digital de Sinal.*

[43] S. Soldado, J. M. N. Vieira, D. F. Albuquerque, and T. Monteiro. Controlling the reconstruction error in Hybrid Filter Banks. *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*, pages 46–50, June 2011.

[44] P. Vaidyanathan. *Multirate Systems And Filter Banks.*

[45] P. Vaidyanathan. Multirate Digital Filter, Filter Banks, Polyphase Networks, and Applications: A Tutorial. *Proceedings of the IEEE, VOL. 78, NO. 1*, 1990.

[46] R. K. Rao Yarlagadda. *Analog and Digital Signals and Systems.* Springer US, Boston, MA, 2010.